United States Army Signal Center, Fort Gordon, Georgia
Leader College of Information Technology

Technical Costs of Implementing a Virtual Private Network

Functional Area 24 Telecommunications Systems Engineer Course, Class 01-02

LTC George B. Hull, MAJ Kenneth Cypher, MAJ Steven Rehn

george.hull@us.army.mil, kenneth.cypher@us.army.mil steven-rehn@us.army.mil

28 May 2002

(FINAL DRAFT)

Table of Contents

Table of Figures

Technical Costs of Implementing a Virtual Private Network

LTC George B. Hull, MAJ Kenneth Cypher, MAJ Steven Rehn

george.hull@us.army.mil, kenneth.cypher@us.army.mil, steven-rehn@us.army.mil

1 INTRODUCTION

Virtual Private Networks (VPN) have spread rapidly both in industry and government as a viable alternative to expensive, leased lines offering fixed bandwidth and a means to provide data confidentiality. Most people today associate VPNs with data networks. However, the first VPNs were not data, but voice. Those first Virtual Private Networks were built by AT&T in order to extend a company's private phone network over public lines to remote locations. Voice VPNs allowed companies to tie together their various facilities across the company with a common numbering plan and more importantly reduced long distance toll rates. The combination of 3 to 5 digit extensions, lower billing rates and the centralization of services such as voice mail accessible from any corporate location helped the voice VPN to spread rapidly in corporate America [1].

As enterprise, wide area data networks developed, the links were typically built across dedicated lines leased from the telephone company. As networks grew in complexity and size, the dedicated leased line solution did not scale well and became a very expensive solution. X.25 packet switching and its successor Frame Relay provided the first alternatives to private leased lines. Telephone service providers built public switched data networks and then sold access to the service to enterprise customers. These technologies allowed the service providers to offer customers "virtual circuits" across their public networks. The customer shared the bandwidth of the public switched data network with other customers. The advantage to the corporate customer was that it no longer bore the full cost of the developing the network between its various sites. Now the customer bought access to a public data network and the service provider provisioned and operated that network.

The drive towards Virtual Private Networks in the data networking world had many of the same precursors seen earlier in the voice market. Enterprise customers sought to tie together geographically dispersed operations with coherent data networks while minimizing costs. In the most general sense a VPN provides a private communications path (a logical or virtual path) between two points across a shared network. Virtual Private Networks can be implemented in many different ways using combinations of a variety of different protocols. Many different technologies have been proposed by vendors to provide VPN capabilities. Various vendors have offered solutions built around frame relay, ATM and IP. These solutions may or may not authenticate users and may or may not encrypt data. For the purpose of this study, a VPN is defined as a trusted network established on an as needed

basis that ensures data confidentiality, integrity, and availability on a transmission path that transcends a non-trusted network. For Army enterprise networks, the attraction of VPNs has had perhaps less to do with economics than with the security advantages that properly implemented VPNs offer to using organizations.

1.1 The Army and Data Confidentiality

Army organizations create and transmit unclassified but sensitive data across computer networks every day. This data classified by DoD 5200.28 as Sensitive But Unclassified (SBU) is defined as information which the loss of, misuse of, unauthorized access to, or modification of might adversely affect U.S. national interest, the conduct of DoD programs, or the privacy of DoD personnel. Examples of the type of data that may be considered to fall into the SBU category include [2]:

- Financial
- Contracting
- Procurement sensitive
- Private corporation proprietary
- Personnel management
- Medical / Health
- Privacy Act information
- For Official Use Only information
- Other Mission Support sensitive data

SBU data may require special handling to preserve privacy or confidentiality of the information. In some cases this information, such as that that involves investigations, Inspector General actions, medical and health information related to patients of procurement sensitive information must be protected to prevent dissemination beyond those with official duties which require access to the information.

This data will be created by variety of different applications and transit a host of different networks. There are a number of public protocols and DoD systems that may meet users needs for data security that will not incur the cost or complexity of establishing and operating a VPN. Before settling on VPNs as a solution to their networking needs, Army information systems managers should analyze their requirements to ensure that other available solutions cannot meet their requirements. VPNs are but one solution to data security and privacy. Some other potential solutions include [3]:

- DoD PKI-enabled Medium Grade Service Messaging
- Secure Socket Layers (SSL) / Hypertext Transport Protocol Secure (HTTPS)
- Transport Layer Security (TLS)
- Kerberos Authentication
- File encryption
- Secure TELNET (STEL)
- National Security Agency Remote Access Security Program (RASP)

1.2 Thesis

This paper will examine the question of how Virtual Private Networks impact network performance when implemented according to DoD guidance.

1.3 Scope

This study is limited to the consideration of technical implementation issues for Virtual Private Networks that carry sensitive but unclassified data on Army NIPR networks. The test network is limited to one constructed of commonly available routers found on most Army installations. The study does not address network performance when using specialized VPN appliances. This study will focus on performance characteristics for selected VPN implementations. This study will not discuss policy issues such as what data should travel on VPNs, nor the relationship between VPNs and the Top Level Architecture, nor will it address specific VPN products.

2 NSA GUIDANCE CONCERNING VPN'S

Before a VPN can be implemented on an Army Network it must meet minimum-security requirements within acceptable risk levels. The process of accrediting a system is outlined in DoD Directive 5200.40, *DoD Information Technology Security Certification and Accreditation Process (DITSCAP)*. Standards used to accredit systems are developed according to the *Common Criteria for Information Technology Security Evaluation.* The Common Criteria is a standard that establishes a common set of requirements for the security functions of IT products and systems and for assurance measures applied to them during a security evaluation [4]. The Common Criteria define the concept of a Protection Profile — an implementation-independent set of security requirements for a category of software that meet specific consumer needs.

The National Security Agency (NSA) defined a Protection Profile for Virtual Private Networks which defines a set of implementation-independent set of security requirements for VPNs in two broad categories: (1) Security Objectives and (2) Functional and Assurance Requirements. VPNs implemented with the DoD must meet these requirements when processing sensitive but unclassified information.

In developing the Protection Profile the NSA did not base it on any specific VPN product but rather on their goals for the technology as a whole. The NSA noted that [5]

> VPNs use security mechanisms to effectively create a private network across a shared (usually public) communications backbone connecting distributed elements or members of a single organization. The interconnecting communications backbone may consist of leased lines, dial-up service, packet and cell switched connection-oriented networks, and/or routed connectionless networks. Also, VPNs are useful in restricting distribution among subsets of the organization at large. This type of nested VPN implementation is commonly referred to as a Community of Interest (COI) within an organization. Typically, VPNs may be utilized to securely communicate between:
>
> • Site-to-site infrastructures across a public communications backbone. This may include Metropolitan Area Networks (MANs) and Building or Base Area Networks

(BANs);
• Local Area Network (LAN)-to-LAN sub-nets operating across a network that services
other entities outside the VPN community;
• Host-to-host workstations across a shared network or sub-net.

2.1 VPN Security Objectives

The NSA's *A Goal VPN Protection Profile for Protecting Sensitive* Information summarizes the VPN Security objectives as shown below [6]:

- Provide confidentiality and integrity protection for unclassified data and user identity as the data moves from an unclassified, sensitive environment through a shared communications backbone to another unclassified, sensitive environment;
- Remove confidentiality protections from peer devices, verify integrity of data between peer VPN devices, and remove integrity mechanisms from the protected (unclassified), sensitive data as it moves from the shared network environment to the receiving sensitive environment;
- Provide mechanisms to restrict the use of the VPN device to Authorized Users, administrators and devices within an operational user site (as identified by IP addresses and passwords);
- Provide authentication mechanisms which restrict the receiving VPN device to Process only information generated by selected VPN peers;
- Provide a limited auditing and alarming capability to record and report VPN related security events (e.g. security connection establishment/termination, failures, and errors);
- Provide local and remote interfaces for VPN administration;
- Support standards-based network operations.

2.2 Functional and Security Assurance Requirements

The U.S. Army Information Systems Engineering Command (USAISEC) summarized the key requirements of VPNs as stated in the NSA Protection Profile. USAISEC notes that the key requirements include confidentiality, data origin authentication, connectionless integrity, protection from data replay attacks, and limited traffic flow security. Specific requirements include[7]:

- Primary security protocol is IPsec (Group 2), RFC number 2401.
- Key generation IAW FIPS 140-2, Level 2
- Key distribution IAW DoD medium assurance PKI for public key distribution using Class 4, X.509 v3 certificates and with hardware tokens for protection of private keys that meet DoD PKI roadmap and FIPS 171 Key Management using American National Standards Institute (ANSI) x9.17.
- Destruction of plaintext cryptographic keys and other unprotected critical security parameters performed within the device IAW FIPS 140-2, Level 2.

- Data encryption is IAW criteria established for either Triple Data Encryption Standard (3DES) IAW FIPS 46-3, SKIPJACK IAW FIPS 140-2, Level 2, or the Advanced Encryption Standard (AES) FIPS 140-2, level 2.
- Signature functions IAW Rivest-Shamir-Adleman (RSA) per PKCS-1 that meets appropriate ANSI standards.
- Data Hashing functions performed IAW Secure Hash Algorithm (SHA-1) meeting FIPS 180-1.
- Key Exchange functions performed IAW Diffie-Hellman Algorithm meeting RFC 2401 for IPSEC (mode 5) and RFC 2409 for Internet Key Exchange, main mode.
- Common Criteria Evaluated Assurance Level 3 (EAL 3)

3 VPN BACKGROUND

3.1 General Architecture

Virtual Private Networks may be viewed through two lenses. The first view defines the protocols used in implementing the VPN. The second view defines the physical topology of the VPN implementation. When considering protocols, designers of VPNs must consider choices from among protocols that deal with tunnels, security and authentication. When considering the physical topology, VPN designers must consider where the trusted endpoints of the VPN will be.

3.1.1 Protocol Overview

Tunneling is the encapsulation of packets or frames inside other packets or frames. Industry uses a number of other protocols besides the IPSec Suite to build VPNs tunnels. These include Layer 2 protocols such as Point to Point Tunneling Protocol (PPTP), Layer 2 Forwarding (L2F) and a protocol which is a combination of the best of both PPTP and L2F — Layer 2 Tunneling Protocol. PPTP, L2F and L2TP are specifically designed to tunnel Point to Point Protocol (PPP) frames through an IP network. Some Layer 3 protocols such as Multiprotocol Label Switching (MPLS) have also been advanced as VPN candidates. Some vendors even market Frame Relay and Asynchronous Transfer Mode (ATM) VPN solutions based on their ability to tunnel TCP/IP and other transport protocols across their networks. Tunneling in and of itself does not secure data from observation, nor does it prevent various other attacks such as IP spoofing, man-in –the middle attacks, replay attacks, etc.

To counter these and other attacks, VPN designers must include security and authentication measures. Security assures data confidentiality. It is implemented through either symmetric encryption algorithms such as Triple DES (3DES) or the Advanced Encryption standard (AES) or through public key asymmetric algorithms such as Rivest-Shamir-Adleman (RSA). Every packet that passes through a VPN tunnel must be encrypted and decrypted, thus efficient algorithms implemented in fast hardware are necessary to provide good network performance.

Authentication assures the integrity of the data packet. TCP/IP includes a checksum as part of the both the IP and the TCP header. This mechanism is designed to detect network transmission errors and is insufficient to protect a data packet against an active attack. An adversary who intercepts the data packet could change the data, recompute the checksum insert it into the header and reinsert the data packet into the network. To maintain the integrity of the data packet requires the use of a Message Authentication Code such as the Secure Hash Algoritm-1 (SHA-1). SHA-1 uses a secure key and computes a one-way hash of the portions of the packet which are be protected. The hash results in a numerical value which is attached to the packet. The authorized receiver of the message shares the SHA-1 key and uses it to calculate the data packet's hash value. If the receiver's calculated value is equal to the value appended to the data packet then the data's integrity has been preserved in transmission.

The NSA has directed the use of the IPSec suite of protocols for developing VPN implementations within DoD. Though IPSec is often discussed as if it were a single protocol it in fact a family of protocols defined and discussed in approximately forty Internet Engineering Task Force (IETF) Requests for Comment (RFC's) [8]. IPSec addresses tunneling, security and authentication. IPSec is a rich suite of protocols which may be combined in many ways to achieve a VPN designer's goals. Its key features will be discussed below. The technical costs of several of the main IPSec implementation options will be discussed in Section 4.

3.1.2 Topology Overview

The VPN is a communications network, established for private use, on top of a shared public network. All VPNs have a start and end point. These are commonly referred to as the trusted endpoints of the VPN. Virtual Private Networks may be categorized by where these trusted endpoints lay in relation to parts of the physical network. Though the three categories defined below are not the only way to describe the topology of a VPN, it is sufficient for this paper.

3.1.2.1 Site-to-Site

Site-to-site VPNs connect geographically dispersed locations through a wide area network. Within the Army each installation would constitute a site connected by the NIPRnet.

3.1.2.2 LAN-to-LAN

LAN-to-LAN VPNs connect separate parts of an organization together across a shared network. This type of VPN could connect two LAN segments on the same installation or it could connect a number of organizations on several installations which wish to operate as if they were on a common network. In LAN-to-LAN VPNs, the traffic for a particular LAN segment is sent through a VPN gateway and then transferred across the public shared network to a VPN gateway attached to another LAN segment.

3.1.2.3 Host-to-Host

Host-to-host VPNs connect workstations and servers across a shared public network. This type of VPN directly connects one computer to another. This type of VPN might be used to connect client workstation on one installation to servers at a different installation to access records that must be kept confidential.

3.2 IPSEC

RFC 2401, Security Architecture for IP states that IPSec is designed to provide interoperable, cryptographically-based security for IPv4 and IPv6. The architecture defines security services for IP to include access control, connectionless integrity, data origin authentication, protection against replays, encryption and limited traffic flow confidentiality. IPSec functions at the IP layer [9].

Document	Title
RFC 2401	Security Architecture for the Internet Protocol
RFC 2402	IP Authentication Header
RFC 2403	The Use of HMAC-MD5-96 within ESP and AH
RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH
RFC 2405	The ESP DES-CBC Cipher Algorithm with Explicit IV
RFC 2406	IP Encapsulating Security Payload (ESP)
RFC 2407	The Internet IP Security Domain of Interpretation for ISAKMP
RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)
RFC 2409	The Internet Key Exchange (IKE)
RFC 2410	The NULL Encryption Algorithm and its Use with IPsec
RFC 2411	IP Security Document Roadmap
RFC 2412	The OAKLEY Key Determination Protocol

Table 3.1 Main IPSec Standards Documents

Though there are many RFC's associated with IPSec, including the twelve main RFC's cited in Table 3.1, IPSec uses four main protocols to provide security — the Authentication Header (See RFC 2402) , the Encapsulating Security Payload (RFC 2406), the Internet Security Association and Key Management Protocol (RFC 2408) and the Internet Key Exchange (RFC 2409). The Authentication Header (AH) provides connectionless integrity, data origin authentication, and an optional anti-replay service. The Encapsulating Security Payload (ESP) provides data confidentiality (encryption), limited traffic flow confidentiality, connectionless integrity, data origin authentication, and anti-replay service. The Internet Security Association and Key management protocol defines procedures and establishes packet formats to negotiate, establish, modify, and delete Security Associations. The Internet Key Exchange (IKE) is used to negotiate the cryptographic algorithm which will be used and to place the necessary cryptographic key at both ends of the tunnel being established.

AH and ESP may be applied alone or in combination with one another. Each protocol supports two modes: transport mode and tunnel mode. In transport mode, the protocols provide security services primarily to the upper layer protocols. In tunnel mode, security

services are applied to tunneled IP packets. IPSec allows the administrator to control the granularity of service. For example an administrator can configure IPSec service to provide a single tunnel to carry all traffic between two gateways (site-to-site). Alternatively, the administrator could configure the service to provide a separate tunnel for every TCP connection between each pair of hosts communicating across the gateway.

3.2.1 Security Associations

Security Associations (SA) are a fundamental component of IPSec. Both Authentication Header protocol and Encapsulating Security Payload protocol use SAs. A key function of the ISAKMP Protocol is the establishment and maintenance of Security Associations. RFC 2401 defines a security association as a simplex connection that affords security services to the traffic carried by the SA [10]. An SA defines the kinds of security measures that should be applied to datagrams based on who is sending the datagram, where it is going and what type of payload it is carrying. Security services are provided to an SA by AH or ESP but not both. If both the AH and ESP protocols are to be used then two SA must be defined. To secure bidirectional (duplex) traffic, SAs are needed for each direction.

A Security Association is uniquely identified by three items: (1) a Security Parameter Index; (2) a destination IP address; and (3) a Security Protocol (AH or ESP) identifier. The Security Parameter Index (SPI) is a 32-bit number usually chosen by the destination endpoint. The SPI has local significance only with the destination endpoint. The Security protocol identifier will either be ESP(50) or AH(51).

RFC 2401 defines two types of SAs: transport mode and tunnel mode. A transport SA is a security association between two hosts. A tunnel mode SA is applied to an IP tunnel. Whenever either end of a security association is a security gateway, the SA must be tunnel mode.

3.2.1.1 Combining Security Associations

The IP datagrams transmitted over a single SA are protected by one security protocol, AH or ESP, but not both. In many cases a security administrator may want to apply more than one security protocol to a traffic stream. This can be accomplished by creating a Security Association bundle. The SAs comprising a bundle may terminate at different endpoints. For example one SA may extend from a host to a security gateway and a second nested SA may extend from that gateway to another security gateway at the destination. Security associations may be bundled in two ways [11]:

(1) *Transport Adjacency* — refers to applying more than one security protocol to the same IP datagram, without invoking tunneling. The most likely example is to have the inner SA apply ESP without its authentication option and the outer SA apply AH to authenticate the entire IP datagram.

(2) *Iterated Tunneling* — refers to the application of multiple layers of security protocols effected through IP tunneling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IPSec site along the path. In an

iterated tunnel for example a host-to-host tunnel can be tunneled through a gateway-to-gateway tunnel. In iterated tunneling any number of nested tunnels may be used.

3.2.1.2 Security Databases

Two databases are associated with each IPSec node. The first, the Security Association Database contains parameters that are associated with each active security association. The second, the Security Policy Database specifies the policies that determine the disposition of all IP traffic in-process or out-process from a host or security gateway. Each interface for which IPSec is defined requires separate in-process and out-process databases.

Yuan [12] summarizes the key fields of the *Security Association Database* as including:

- The Security Parameter Index (SPI)
- The protocol to be used for the security association (AH or ESP)
- The mode in which the protocol is operated (tunnel or transport)
- The sequence number counter
- The anti-replay window
- The path maximum transmission unit
- The source and destination IP addresses of the security association
- The authentication algorithm to be used and the authentication key
- The encryption algorithm and the encryption key
- The lifetimes for the authentication and encryption keys
- The lifetime of the security association

For an in-process IP packet, the appropriate SA is found in the security association database by matching three values with information in the IP datagram header: the destination IP

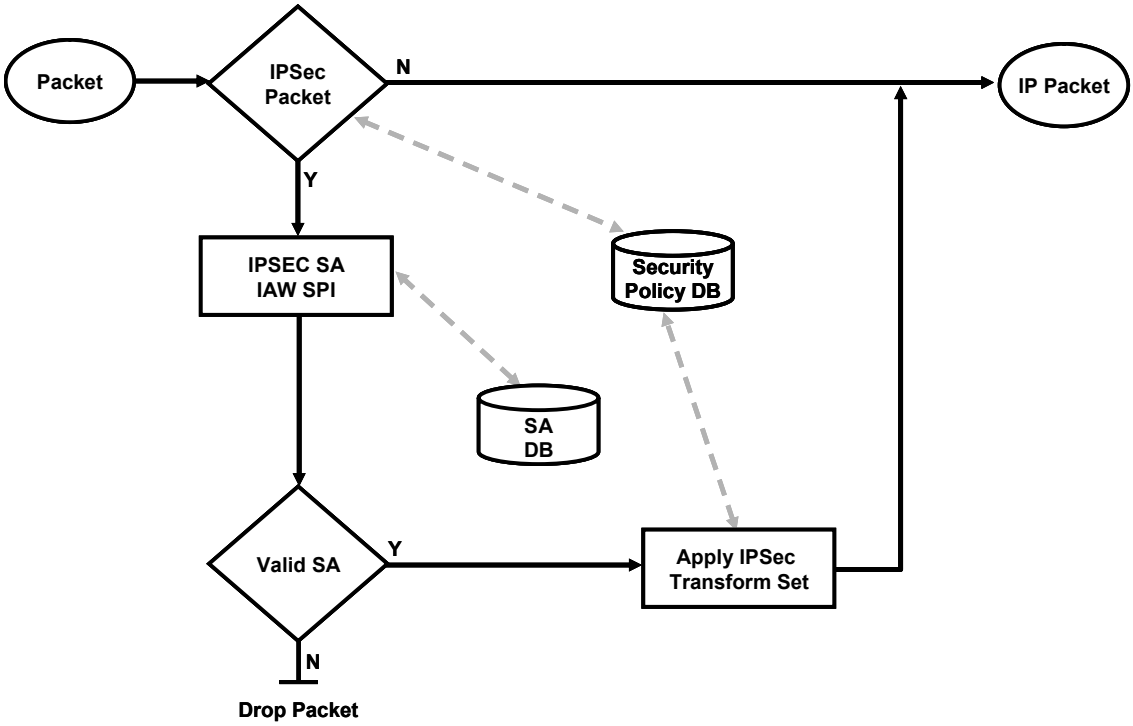


Figure 3.1 In-process Flow

address, the IPSec protocol type and the SPI. If an SA is defined in the security association database for then the IP packet is processed according to the parameters specified in the database entry. If an SA is not found the packet is discarded.

For out-process IP datagram processing, the datagram is first processed in accordance with security policy database requirements (discussed below), then the security association database is searched to see if an SA is already established. If an SA is established its parameters are used to process the datagram. If no SA is established a new SA is negotiated for the datagram with the receiving end. The new SA is then stored in the security association database.

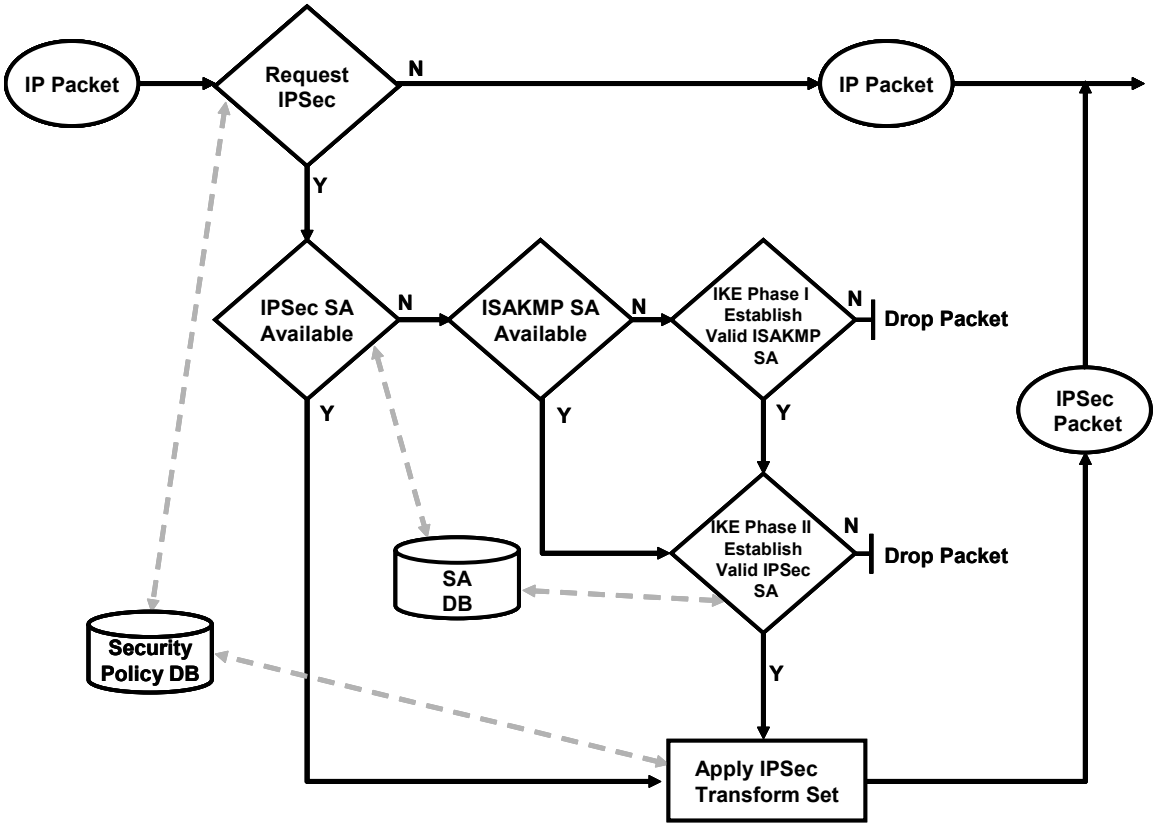


Figure 3.2 Out-process Flow

The *Security Policy Database* specifies what services are to be offered to IP datagrams and in what manner. The security policy database is used to specify general security rules, or policies, that should be applied to IP datagrams. It is an ordered list of security policies that are applied in sequence to IP datagrams as they are processed through an IPsec interface. The policies are specified through two components: a selector and an action. The selector's that IPsec currently uses are [13]:

- Destination IP address
- Source IP address
- Transport layer protocol

- System name (fully qualified DNS, or e-mail address, or X.500 DN
- User ID (fully qualified DNS user or an X.500 DN

The selector can be thought of as a screening criterion. It might be a source IP address or an e-mail address or a distinguished name. Selectors based on information that is not available in an IP header, for example a distinguished name, are translated into IP addresses during the IKE negotiation process. Selectors can be combined using Boolean operators and wildcards to create complex expressions. The action is the action to be taken when the selector criteria is matched. For instance, an action might be to apply IPSec with ESP where 3DES is the specified encryption algorithm. It is possible for an IP datagram to match more than one entry in the security policy database. However, the first match in the policy database will be the one applied. When no match is found the default policy is applied. The default policy is usually to discard the datagram.

The security policy database must be consulted for out-process and in-process traffic for every packet (to include non-IPSec packets). For any out-process or in-process datagram, three choices are possible: discard the datagram, bypass IPSec, or apply IPSec. When an in-process IP datagram arrives at an IPSec enabled interface, IPSec first searched the security association database for the appropriate SA. When an SA is found, the system initiates the security association database instructions and then conducts security policy database processing. For out-process datagrams security policy database processing is done first. If the matching security policy database entry specifies IPSec processing, then the security association database is searched to see if an SA is established. If an SA is established then the packet is processed according to the SA. If an SA has not been established, one is negotiated.

3.2.2 IP Authentication Header

The IP protocol is inherently insecure. As discussed above, the checksum field was designed to identify transmission errors and offers no assurance that the data packet was not modified enroute between the sender and the receiver. The IP Authentication Header (AH) provides per packet connectionless integrity and data origin authentication for IP datagrams. AH provides strong cryptographic authentication protection for the all the fields of the IP datagram that do not change.[1] This ensures that a packet cannot be tampered with while in transit without detection.

To provide authentication of a data packet, the sender computes an integrity check value, places it in the authentication data field and sends it with the AH. If during AH processing the datagram size exceeds the maximum transmission unit size allowed, then the datagram is fragmented. Upon receipt, the fragments are reassembled prior to further AH processing. If all fragments are not received, AH discards the datagram. The integrity check value is a keyed hash value over all the fields of the IP datagram except those parts of the header which change as the packet traverses the network. The sender and receiver negotiate a secret key during the establishment of the security association. Upon receipt of a packet the receiver

[1] Some fields in an IP Header are mutable. They change as the IP datagram travels through the network. The Time to Live (TTL), Type of Service, Flags, Header Checksum, Options and Fragment Offset are the fields that change.

calculates the hash value and compares it to the integrity check value contained in the authentication data field of the AH. If they match the IP datagram is assumed to be good. If the packet cannot be authenticated it is discarded.

3.2.2.1 The AH Header

The AH header consists of six fields; five of fixed length and one — the Authentication Data — of variable length (SHA-1 is 12 bytes, creating a standard 24 byte header). Each field is defined in RFC 2402 [14]:

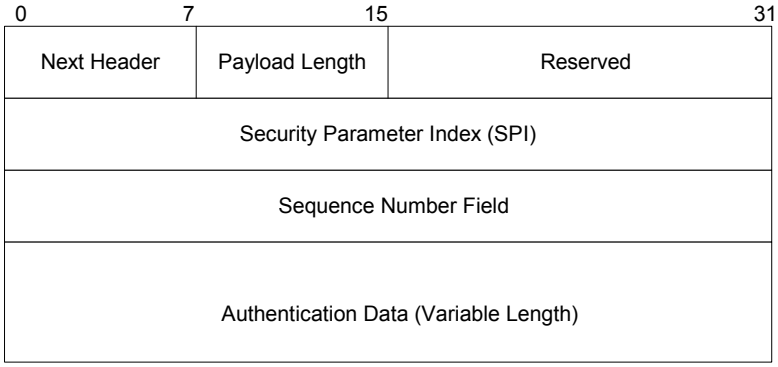


Figure 3.3 Authentication Header

- *Next Header* — an 8 bit field that identifies the type of the next payload after the Authentication Header. The value of this field is chosen from the set of IP Protocol Numbers defined in the most recent "Assigned Numbers" (STD-2) RFC from the Internet Assigned Numbers Authority.
- *Payload Length* — an 8 bit field that specifies the length of the AH in 32 bit words (4-byte units) minus "2".
- *Reserved* — a 16 bit field reserved for future use. It must be set to zero.
- *Security Parameters Index* — The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (AH), uniquely identifies the Security Association for the datagram. The SPI is normally selected by the destination system upon establishment of a Security Association.
- *Sequence Number* — an unsigned 32-bit number that contains a monotonically increasing counter value (sequence number). The sender must always send the field, but the receiver does not have to act on it. The sender and receiver's counters are initialized to zero when a new Security Association (SA) is established. The transmitted number must never be allowed to cycle, thus the sender and receiver's counter must be reset (by establishing a new Security Association and thus a new key) prior to transmission of 2^{32d} packet on an SA.
- *Authentication Data* — a variable length field that contains the Integrity Check Value for the packet. The field must be an integral multiple of 32 bits. The field may include padding to round the data out to an even 32 bit integral. IPsec mandates that HMAC-MD5 and HMAC-SHA-1 be implemented in all IPsec implementations.

3.2.2.2 AH Transport Mode

In AH transport mode, the original IP header is retained as the header of the new packet. The authentication header is inserted after the IP header and before any transport protocol header or other IPsec headers. AH authenticates the entire IP header minus the mutable fields.

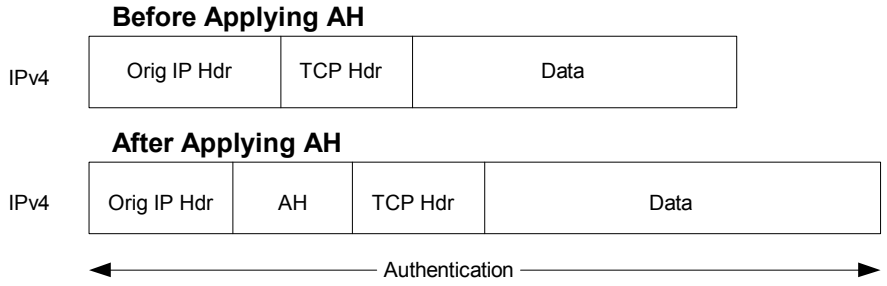
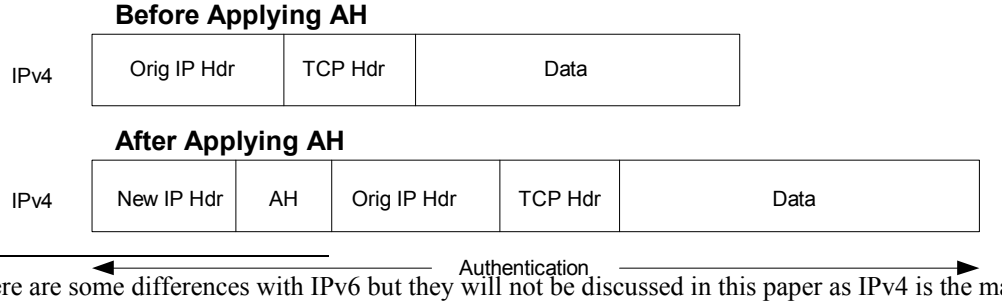


Figure 3.4 Inserting the AH header in Transport Mode[2]

Transport mode adds only 24 extra bytes of overhead utilizing SHA-1 to the original IP packet. However, it is only practical to implement it between end hosts. Most enterprise security environments will encounter problems using AH in transport mode. Davis [15] points out that in organizations which use non-routable private addresses on their internal network and then pass the packets through a Network Address Translator (NAT) at the organization gateway with the internet can encounter problems with the use of AH. NAT replaces the internal address with a routable address at the organization's network gateway. If AH is applied to the datagram prior to NAT processing, the AH check at the receiving end will fail due to the replacement of the IP address on the datagram. Similar problems can occur with hosts that have routable addresses but lie behind security firewalls. In these scenarios, if AH transport mode is to be used it must be enabled on the security gateways and the Security Associations must be established between gateways.

3.2.2.3 AH Tunnel Mode

In AH tunnel mode the original IP header is replaced with a new one and the original IP



[2] There are some differences with IPv6 but they will not be discussed in this paper as IPv4 is the main protocol in use. See RFC 2402 for details on how AH affects IPv6.

Figure 3.5 Inserting the AH Header in Tunnel Mode

datagram is encapsulated in a new IP datagram. The AH header is inserted between the new IP header and the original IP header. Authentication is applied as before to the entire datagram minus the mutable fields of the new IP header. The original IP packet is maintained completely intact and thus the authentication covers every field in the original IP datagram. The limitations discussed concerning AH in transport mode are applicable in tunnel mode as well. Tunnel mode adds 44 extra bytes of overhead utilizing SHA-1 to the original IP packet.

3.2.3 Encapsulating Security Payload

The Encapsulating Security Payload (ESP) protocol is designed to provide a security services., such as, data confidentiality (encryption), data origin authentication, connectionless integrity, anti-replay service, and limited traffic flow confidentiality. Encryption is only provided by symmetric-key encryption algorithms. The NSA VPN Protection Profile specifies 3DES, Skipjack or AES

3.2.3.1 The ESP Packet Structure

The ESP packet structure is inserted after the IP header and before the upper layer protocol header (transport mode) or before an encapsulated IP header (tunnel mode). The ESP packet consists of four fixed-length fields and three variable-length fields. The header is comprised of the SPI and the Sequence Number Field. The trailer starts with the Pad length field. The minimum overhead added to the IP packet is 22 bytes without AH and 34 bytes will AH. RFC 2406 defines the fields as [16]:

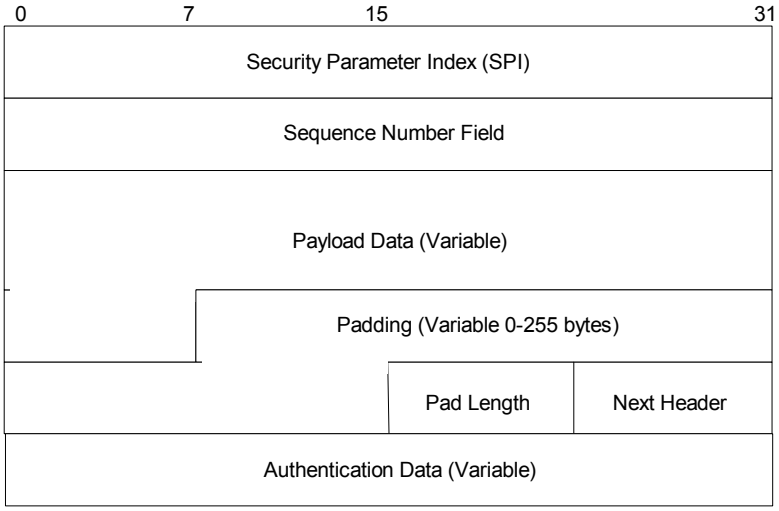


Figure 3.6 The Encapsulating Security Payload Header

- *Security Parameter Index* — The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (ESP), uniquely identifies the Security Association for the datagram. The SPI is normally selected by the destination system upon establishment of a Security Association.

- *Sequence Number Field* — an unsigned 32-bit number that contains a monotonically increasing counter value (sequence number). The sender must always send the field, but the receiver does not have to act on it. The sender and receiver's counters are initialized to zero when a new Security Association (SA) is established. The transmitted number must never be allowed to cycle, thus the sender and receiver's counter must be reset (by establishing a new Security Association and thus a new key) prior to transmission of 2^{32d} packet on an SA.
- *Payload Data* — Payload data is a variable length field containing the data described by the *Next Header* field. The payload data field contains is an integral number of bytes in length. If the encryption algorithm requires an Initialization Vector, then it may be sent as part of the payload data, but the RFC implementing the encryption algorithm must describe how such a vector is placed into the payload and how it is recovered.
- *Padding* — A variable length field between 0 and 255 bytes. The padding field is used to support the requirements of some encryption algorithms to have data fields of some prescribed length. For example some encryption algorithms require the plaintext to be the multiple of some number of bytes, such as the block size of a block cipher. Padding may also be needed to insure that the resulting cipher text terminates on a four-byte boundary in order to line up the start of the *Pad Length* and *Next Header* fields on a four-byte boundary as required. Padding can also be used to conceal the actual size of the payload, making traffic analysis more difficult.
- *Pad Length* — This 8-bit field indicates the number of bytes in length that the *Padding* field occupies. This field will contain a number between 0 and 255.) indicates no padding was added to the payload.
- *Next Header* — An 8-bit field that indicates the type of data contained in the *Payload Data* field. The value of this field is chosen from among the IP Protocol numbers defined in the most recent, "Assigned Numbers," [STD 2] RFC from the Internet Assigned Numbers Authority (IANA).
- *Authentication Header* — This field contains an Integrity Check Value computed over the ESP packet minus the authentication data. The length of the field is dependent on the authentication function (SHA-1, MD5 etc.) selected. The Authentication Data field is only includes if the ESP authentication service is specified in the relevant SA.

3.2.3.2 Encapsulating Security Payload Transport Mode

In transport mode, ESP is inserted after the IP header, but before any upper layer protocol headers or IPSEC (i.e. AH) protocols that may already have been applied .The ESP header fields contain the SPI and sequence fields, while the ESP trailer fields contain the

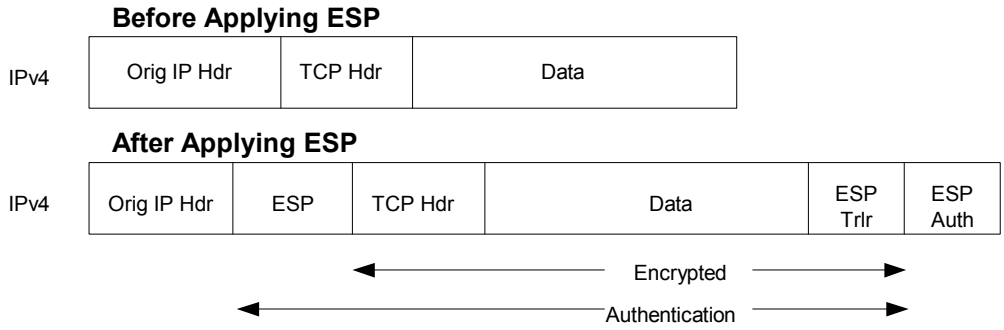


Figure 3.7 Inserting the ESP Header in Transport Mode

padding, pad length and next header fields. Davis [17] states that the ESP header and trailer fields are not encrypted as information contained in them is needed at the destination node for packet processing.

Note that the ESP encryption service does not include the original IP header. This means that ESP transport mode offers no traffic flow confidentiality. It also means that datagrams in ESP transport mode can negotiate firewalls and network address translation servers without the problems encountered with the AH header. However, this flexibility comes at a price. Because the original IP header is not authenticated, the datagram could be intercepted, changed or otherwise tampered with during transmission and the receiver would have no method of detecting this. Thus, ESP transport mode authentication service offers less security than that provided by AH transport mode.

3.2.3.3 ESP Tunnel Mode

In tunnel mode ESP is inserted before the original IP header and a new IP header is inserted before the ESP header. In tunnel mode ESP provides encryption and authentication to the entire IP datagram.

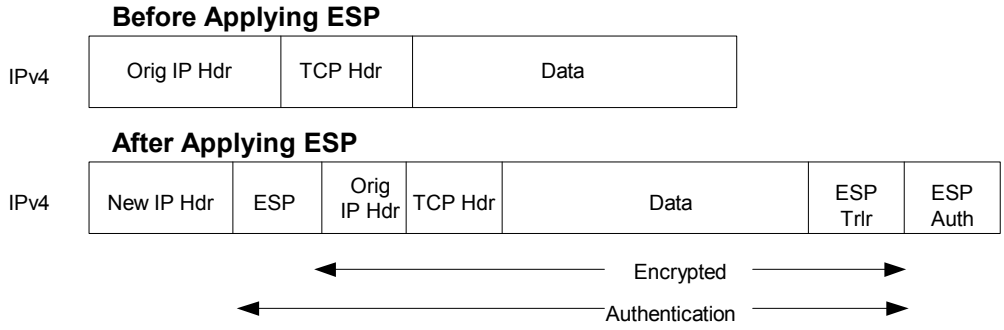


Figure 3.9 Inserting the ESP Header in Tunnel Mode

The minimum overhead added to the IP packet is 42 bytes without AH and 54 bytes with authentication. The outer IP header is neither encrypted nor authenticated. If it were encrypted it could not be routed through the network. If the new IP header was authenticated, it would not pass its validity test at the destination if it had to pass through firewalls or network address translation servers that changed the IP address at the network gateway. When ESP tunnel mode is implemented at security gateways it does offer traffic flow confidentiality service as the original IP headers with the true source and destination IP addressees are encrypted.

3.2.4 Internet Security Association and Key Management Protocol

The Internet Security Association and Key Management Protocol (ISAKMP) defines procedures and packet formats to establish, negotiate, modify and delete Security Associations. ISAKMP was designed to apply to a range of protocols and thus is not bound to any specific encryption technique or key exchange algorithm. ISAKMP provides the protocol exchanges to establish a security association between a source and destination computer. RFC 2408 states that an initial protocol exchange allows the two entities to agree on a basic set of security parameters that will be used to secure all future negotiating messages. After the initial security agreements set up a secure messaging procedure, the identity of the negotiating computers must be authenticated, encryption and data authentication algorithms agreed to, and the required keys generated and exchanged. At this point the Security Association can be used for further communication between the two computers[18].

3.2.4.1 The ISAKMP Header

ISAKMP messages consist of a fixed-length header followed by a variable number of payloads. RFC 2408 defines thirteen different payloads. The payloads provide the building blocks for constructing the ISAKMP messages. The minimum bytes added is 40 extra bytes of overhead utilizing the smallest payload list in the next section to the original IP packet. The ISAKMP header is defined as follows [19]:

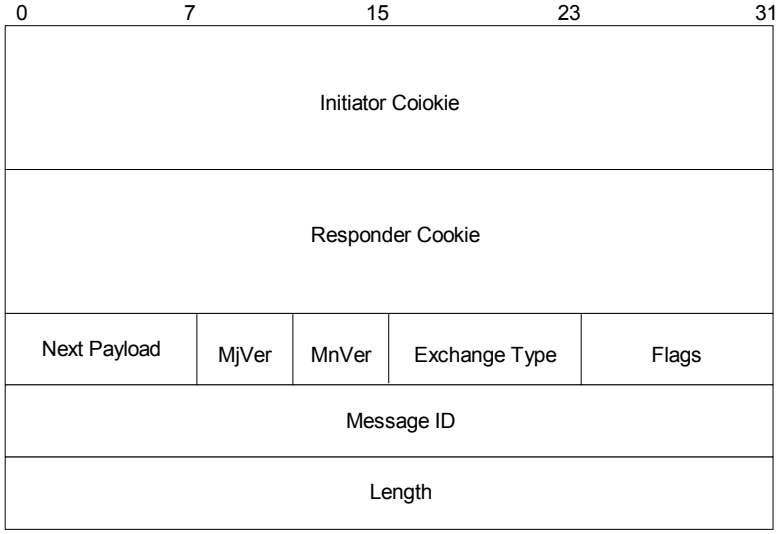


Figure 3.11 ISAKMP Header

- *Initiator Cookie* — an unique 8 byte string that is generated by the ISAKMP message initiator, initiating SA establishment, SA notification, or SA deletion.. It is used to protect the message exchanges from compromise by an attacker that obtains the message, changing information then using it to swamp the victim with Diffie-Hellmen requests from randomly chosen IP addresses. RFC 2408 refers to the Initiator and Responder Cookies as "Anti-Clogging Tokens." If the responder determines that the cookie does not match the one received previously from the initiator, then the responder drops the message without any further processing. This technique helps prevent denial of service attacks.
- *Responder Cookie* — an unique 8 byte string from the entity that is responding to an SA establishment request, SA notification, or SA deletion.
- *Next Payload* — (1 byte) field that indicates the type of the first payload in the message. The possible values are:
  - None 0
  - Security Association 1
  - Proposal 2
  - Transform 3
  - Key Exchange 4
  - Identification 5
  - Certificate 6
  - Certificate Request 7
  - Hash 8
  - Signature 9
  - Nonce 10
  - Notification 11
  - Delete 12
  - Vendor ID 13
  - Reserved 14-127
  - Private Use 128-255
- *Major Version* — (1 byte) Indicates the major version of the ISAKMP protocol in use.
- *Minor Version* — (1 byte) Indicates the minor version of the ISAKMP protocol in use
- *Exchange Type* — (1 byte) Indicates the type of exchange being used. This dictates the message and payload ordering in the ISAKMP exchanges. Values are defined as:
  - None 0
  - Base 1
  - Identity Protection 2
  - Authentication only 3
  - Aggressive 4
  - Informational 5
  - ISAKMP Future Use 6-31
  - DOI Specific Use 32-239

  - ➔ Private Use 240-255
- *Flags* — (1 byte) Indicates specific options that are set for the ISAKMP exchange. Only the first 3 bits of this field are currently used, the last five bits are set to zero before transmission. The first three bits are defined as:
  - ➔ Encryption — If set equal to 1 then all payloads following the header will be encrypted using the encryption algorithm specified in the ISAKMP SA. The ISAKMPO SA is the combination of the Initiator and Responder Cookie. If set equal to 0, then no encryption is applied.
  - ➔ Commit — This bit is used to signal key exchange synchronization. It is used to ensure that encrypted material is not received prior to completion of the SA establishment. When set equal to 1, the entity that did not set the bit must wait until it receives an informational exchange containing a Notify Payload from the entity that set the Commit bit.
  - ➔ Authentication Only — If set equal to one, only authentication services will be applied to the Payload. The Payload will not be encrypted.
- *Message ID* — (4 bytes) Unique message identifier used to identify protocol state during Phase 2 Negotiations. This value is randomly generated by the initiator of the Phase 2 negotiations. During Phase 1 negotiations the value must be set to 0.
- *Length* — (4 bytes) Length of the total message (Header + Payloads) in bytes.

3.2.4.2 The ISAKMP Payloads

All ISAKMP Payloads are preceded by a 4-byte generic header comprised of three fields:

- *Next Payload* — (1 byte) Identifier for the payload type of the next payload in the message. If the current payload in the message is the last in the message, then this field will be 0.
- *Reserved* — Unused, set to 0.
- *Payload Length* — Length in octets of the current payload, including the generic payload header.

Using the generic header payloads may be chained together in a single ISAKMP message. The generic header serves as the clear delineation between subsequent payloads. As mentioned above, ISAKMP defines thirteen message payloads. The message payloads are listed below.[3] The message payloads defined in RFC 2408 are [20]:

- *Security Association Payload*
- *Proposal Payload*
- *Transform Payload*
- *Key Exchange Payload*
- *Identification Payload*
- *Certificate Payload*
- *Certificate Request Payload*
- *Hash Payload*

[3] Details concerning the specific payload fields and their definitions may be found in RFC 2408, p.25-45.

- *Signature Payload*
- *Nonce Payload*
- *Notification Payload*
- *Delete Payload*
- *Vendor ID Payload*

3.2.5 The Internet Key Exchange Protocol

Internet Key Exchange (IKE) is a protocol that is used to negotiate and provide authenticated keying materials in a protected manner for the establishment of security associations. IKE is a hybrid protocol that implements a subset of the Oakley Key Determination Protocol, a subset of the SKEME Secure Key Exchange Mechanism protocol and uses message formats and procedures specified in Internet Security Association and Key Management Protocol. IKE provides keying material for IPSec peers, from which encryption and authentication keys can be generated [21].

IKE presents different exchanges as modes which operate in two phases. Phase 1, Main Mode, is used for establishing and securing the ISAKMP SA for key exchange and authentication of the ISAKMP SA. Phase 2, Quick Mode, is used to negotiate Phase 2 SA under the protection of the negotiated Phase 1 ISAKMP SA. Once the ISAKMP SA is established, the requesting protocol has a protected path for negotiations of its own SA. Phase 2 negotiations are conducted for protocols such as, ESP or AH, that need key material or a parameter negotiation. When the requesting protocol's SA is established, the ISAKMP SA is torn down.

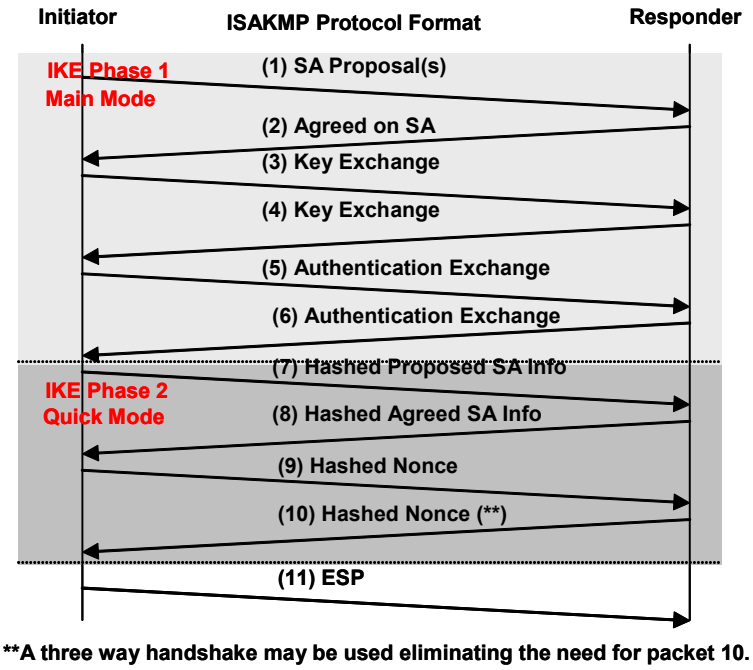


Figure 3.10 IKE Information exchange for SA Establishment

ISAKMP message formats (payloads) are used in quick mode to check the SA status and exchange information on keys for each new file transfer. Since the Phase 1 SA establishment verified the recipient and exchanged keys, the headers of the quick mode packets contain the identifying Digital Signature to notify the recipient which keys to use for decryption. This process reduces the required overhead associated with the initial authentication and key transfer for each transmitted packet.

The details of the payload fields and options vary dramatically and are covered in detail in RFC 2408 "Internet Security Association and Key Management Protocol." All IKE messages have a standard header of 28 bytes plus their payload. The payload may range from 12 bytes for a proposal payload message to several hundred bytes based on the specific payload.

ISAKMP and IKE are used within the IPSec suite to establish security associations. In IPSec, SA's are only removed by three methods. First, they may be manually removed. Second, they may expire at a preset lifetime. Third, the SPI number may exceed 2^{32} and thus start over, initiating a renewing of the SA..

IKE requires that the encryption algorithm, hash algorithm, authentication method, and Diffie-Hellman group to be negotiated as part of the Phase 1 ISAKMP SA establishment. IKE implementations must support DES, MD5, SHA-1, authentication via pre-shared keys, and MODP group number one. IKE recommends support of 3DES for encryption, Tiger for hash, RSA for digital signature standard, and MODP group number 2.

4 EVALUATION OF PROTOCOL OVERHEAD

4.1 Overview and Methodology

This study measured the performance characteristics of selected VPN implementations to determine the technical impact on throughput, response time, processor latency and transaction rate in a DoD complaint (IPSec/3DES/SHA-1) LAN-to-LAN VPN implementation. The intent was not to simulate a VPN in a live network, but to generate statistics in a clean environment where general performance could be measured. The resulting measurements then provide a benchmark to aid in the understanding of performance related impacts of implementing VPNs. Performance impacts under conditions outside this scope of this study must be considered before implementing a VPN solution into a live network. The results reported in this study are not intended as an evaluation of a particular vendor solution.

An isolated network was created to measure and compare non-encrypted, baseline, network characteristics against those same characteristics of an encrypted network utilizing the requirements for a DOD VPN (See Sec 2.2). VPN performance was evaluated using three different methodologies to generate traffic and take measurements:

- Case A: Chariot Application Flow Simulator
- Case B: CMPMetrics for VPNs
- Case C: UDP Flood

Each case utilizes the same basic network structure. The evaluation consists of establishing two sets of VPN LAN-to-LAN peers conforming to DOD VPN requirements[4]. All functions of the VPN are performed on a gateway making the VPN transparent to the end user. Performance characteristics for three different IPSec security associations (SA) are measured for each comparison methodology:

1. Authentication Header(AH) - Transport Mode
2. Encapsulating Security payload (ESP) – Tunnel Mode
3. ESP with Authentication (ESP-A) – Tunnel Mode

In addition, a measurement of performance characteristics is performed on the network without an active IPSEC SA to establish baseline characteristics. An ESP-A SA tunneled within a second ESP-A SA is also evaluated briefly.

For each case examined the following performance characteristics were measured:

1. *Delay:* The amount of processing time required by the device performing VPN functions.

[4] Resource constraints prevented the creation of exactly the conditions specified for a DoD VPN as outlined in Sec 2.2. See Sec 4.2 for a discussion of the limitations.

2. *Throughput:* The amount of data transferred in a specific amount of time. Measured using the following formula:

$$\frac{\left(\frac{\text{Bytes Sent + Bytes Received}}{\text{125,000 bytes per second}}\right)}{\text{Measured Time}}$$

   Where measured time is the sum of the time required to execute a complete transaction from all timing records returned for an endpoint pair. 125,000 bytes is equal to 1MBps (1,000,000 bits/8 bits per byte).

3. *Transaction Rate:* The number of transactions that occur per second. Calculated as follows:

$$\frac{\text{Transaction Count}}{\text{Measured Time}}$$

4. *Response Time:* The amount of time required to complete one transaction; the inverse of Transaction rate. The Cooperative Association for Internet Data Analysis (CADIA) considers response time in excess of 300ms unsatisfactory.[22] Opnix, Inc's Internet Traffic Report indicates the average response time for traffic on the Internet from 25 Apr 02 to 25 May 02 is 177ms globally and 99ms in North America.[23]

$$\frac{\text{Measured Time}}{\text{Transaction Count}}$$

4.1.1 Limitations

In creating the network environment for this study, the following limitations were encountered:

1. DOD PKI was not implemented as an authentication mechanism for the VPN. In lieu of a DOD PKI certificate, a PKI Certificate was generated from standalone Certificate Authority using Microsoft Certificate Server. The key strength was maximized to 2048. The impact on this evaluation is minimal, as the certificate is only used during IKE Phase I authentication during establishment of an ISAKMP SA.
2. The Cisco IOS prevents the establishment of LAN-to-LAN transport mode SAs. The transport mode SAs can only be used in a Host-to-Host implementation.
3. The test network used, lacked sufficient equipment to simulate a wide area network link. Additional links create the likelihood for additional delay and increased response time. This increased time becomes a critical performance factor for applications that are time sensitive and/or use smaller packets, which contain a larger percentage of overhead produced by the VPN.
4. Packets are carried within an Ethernet II frame and are limited to an IP Datagram size of 1500 bytes.

4.1.2 Network Topology.

The equipment and software used to construct the network test environment is listed below:

- 4 x Cisco 3640 Routers (R4700 CPU @ 100 MHz, 32 MB Flash, 64 MB RAM) with IOS 12.2.8 with ENTERPRISE/FW/IDS PLUS IPSEC 3DES feature set (c3640-jk9o3s-mz.122-8.T).
- 3 x Desktop Computers (WIN NT 4.0 build 1381 w/ service pack 6, Intel Pentium 266Mhz, 64 MB RAM)
  - ➔ NetIQ's Performance Endpoint software for Microsoft Windows NT, Windows 2000, and Windows XP. Software agents used by Chariot to simulate, collect, and report information about network transactions for analysis and reporting. (http://www.netiq.com/support/chr/pe.asp)
- 1 x Desktop Computers (WIN NT 4.0 build 1381 w/ service pack 6, Intel Pentium 266MHz, 64 MB RAM)
  - ➔ NetIQ's Performance Endpoint software for Microsoft Windows NT, Windows 2000, and Windows XP
  - ➔ EtherReal Network Analyzer version 0.8.20.0. Protocol Analyzer. (http://www.ethereal.com/)
  - ➔ UDPFlood v2.0 - UDP packet sender utility. (http://www.foundstone.com/knowledge/proddesc/udpflood.html)
- 1 x IBM Think Pad (2611-411) (Win 2k Server, Intel Pentium 233MHz, 164 MB RAM).
  - ➔ NetIQ's Chariot version 4.3. Chariot evaluates the potential performance of networked applications by performing stress tests on network devices utilizing simulated application traffic flows. (http://www.netiq.com/products/chr/default.asp)
  - ➔ EtherReal Network Analyzer version 0.8.20.0
  - ➔ CMPMetrics for VPN Metrics. Measures Throughput and Response time based upon Chariot utilizing modified scripts designed to test VPNs. (http://www.networkcomputing.com/cmpmetrics/)
  - ➔ MS Certificate Server (Windows 2000 Server) with Certificate Services Add-on for Cisco Enrollment Protocol (CEP) (cepsetup.exe) from the Windows 2000 Resource Kit companion CD.

The basic network configuration used for the evaluation during all three cases discussed below is illustrated in Figure 4.1

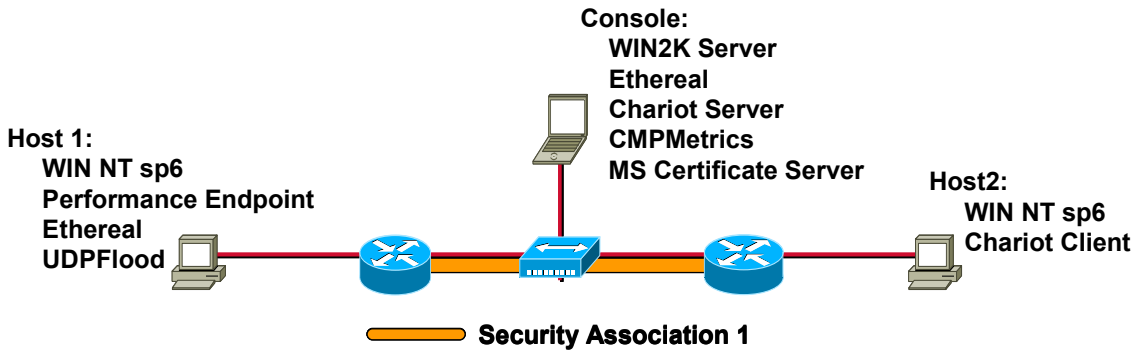


Figure 4.1 VPN Test Network

4.2 Experimental Methodologies

This section presents a discussion of the specific methodology used for each study case and the associated findings. The cases were selected on the ability to evaluate specific criteria. Case A, Chariot Flow Simulator, simulates specific application flow and allows for the evaluation of response time, throughput, transaction rate and time for a given type of application. Case B, CMPMetrics was selected to provide a relative comparison of the performance impact of implementing different IPSec SAs has on throughput, response time and elapsed time, The delay caused by processing an IPSec SA was measured through Case C, UDPFlood.

4.2.1 Case A — Chariot Application Flow Simulator

Utilizing Chariot, a software application that mimics application flows and gathers specific performance characteristics, a series of test were performed to record specific network characteristics to compare the overall network performance under different IPSec SAs to a baseline.

4.2.1.1 Case A — Methodology

The Console computer was configured with the initial test parameters to include execution script. These parameters were sent to Host1 for execution. During execution, Host1 acts as the client computer while Host2 acts as the server. Host1 calculates performance characteristics for each data transfer and transfers the results to the Console.

Basic Methodology: The basic methodology to measure delay, throughput, transaction rate, and response time included:

- *STEP 1:* Establish a baseline with no IPSec SAs. Emulate and measure network performance under normal conditions utilizing Chariot and the selected scripts between Host1 and Host2. Performance measurements are conducted separately for each script selected, with the exception of scripts 6, 7, 8, 9, and 10, which are run simultaneously. Five iterations are performed for each script selected.
- *STEP 2:* Repeat *STEP 1* with an AH-Transport mode IPSec enabled on SA1.
- *STEP 3:* Repeat *STEP 1* with an ESP-Tunnel mode IPSec enabled on SA1.

- *STEP 4:* Repeat *STEP 1* with an ESP-A-Tunnel mode IPSec enabled on SA1.
- *STEP 5:* Compare the results.

Test Conditions. Utilizing the scripts available for Chariot from NetIQ, we selected several to provide a broad spectrum of conditions to evaluate the IPSec SA modes. Selected scripts include, the following as describe by NetIQ [24]:

- *Credit Long.* (Script used: creditl.scr)
  - ➔ *Description.* This script emulates a series of credit approvals. A record is sent from Host1 to Host2. Host2 receives the record and sends back a confirmation. A single connection is established for all transactions. The default record size of 100 bytes was selected.
  - ➔ *Rationale.* Focus on the response time of a high transaction rate database connection.

- *Database inquiry with a Long Connection*. (dbases_long.scr)
  - ➔ *Description.* This script emulates a program on Host1 that requests a record from Host2, receives, updates and returns the record back to Host2. Lastly, Host1 receives a confirmation from Host2 that the update was completed. A single connection is established for all transactions. The size of each transaction was set for 100 bytes with 25 transactions set per second.
  - ➔ *Rationale*. Selected for the complexity of the transaction to simulate database operations over a VPN connection.

- *Receive Email via POP3.* (pop3.scr)
  - ➔ *Description.* This script emulates typical e-mail retrieval from a POP3 server. The size of an e-mail message is set to 1,000 bytes, with a 20-byte reply and 70 byte control flow and uses a newsgroup type email message format. The transaction rate was set to 5 per second.
  - ➔ *Rationale.* Measure the impact of VPN on a POP3 connection to determine the impact on non-local email users.

- *Large File Send with Short Connection*. (filesndshort.scr- Modified).
  - ➔ *Description*. Emulates sending a file from Host1 to Host2 with a return confirmation. A new connection is made for each file transfer. File size set to 1,000,000 bytes.
  - ➔ *Rationale.* Measure the impact of large file transfers.

- *File Send with Short Connection*. (filesndshort.scr.)
  - ➔ *Description*. Emulates sending a file from Host1 to Host2 with a return confirmation. A new connection is made for each file transfer. File size set to 100,000 bytes.
  - ➔ *Rationale*. Measure the impact of file transfers.

- *Small File Send with Short Connection*. (filesndshort.scr- Modified).
  - ➔ *Description*. Emulates sending a small file from Host1 to Host2 with a return confirmation. A new connection is made for each file transfer. File size set to 5000 bytes.

➔ *Rationale*. Measure the impact of small file transfers.

- *Active Directory Login*. (Actdlog.scr).
  - ➔ *Description*. Emulates the data flows generated when a user logs in from a Windows 2000 Professional computer to a Windows 2000 Server computer.
  - ➔ *Rationale*. Measure the response time a single user experiences when attempting to log in to a Domain Controller.

- *Active Directory Replication.* (actdrep.scr).
  - ➔ *Description*. Emulates the replication of a 580,000 bytes directory under Active Directory for Windows 2000.
  - ➔ *Rationale.* Measure the impact on Active Directory and Windows 2000.

- *Exchange Read*. (exchread.scr).
  - ➔ *Description*. Emulates a client, Host1, retrieving email messages from the Exchange Server, Host2. Host1 requests the full list of unread email messages from Host2, who sends the unread email messages to the client. The size of email messages is 2,800 bytes. This variable can be edited to reflect the average size of email message that is to be used in testing..
  - ➔ *Rationale*. Measure the impact on local Email services.

- *Exchange Send*. (exchsend.scr).
  - ➔ *Description*. Emulates the sending of email by a Microsoft Exchange client. Each transaction represents the transfer of an email message from the client, Host1, to the server, Host2, who returns an acknowledgement to Host1. The default for the email message size is 1,488 bytes, which includes 700 bytes of Exchange email control information and 788 bytes of readable text.
  - ➔ *Rationale*. Measure the impact on local Email services.

- *Exchange Receive*. (exchrecv.scr).
  - ➔ *Description*. Emulates a Microsoft Exchange client periodically receiving notification of new email messages. The client, Host1, requests the list of unread email "headers" (sender and subject) from the server, Host2. Host2 then sends the list of unread email "headers" to the client. This script does not include an 8-byte UDP message, which the mail server sends to the client to inform the client that there is a new message on the server. The default script uses one unread message with a typical header size of 816 bytes.
  - ➔ *Rationale*. Measure the impact on local Email services.

4.2.1.2 Case A — Findings.

Performance was adversely impacted on all simulated applications when a IPSec SA was applied with the exception of POP3. Applications composed of small file size transactions and numerous queries suffered the greatest degradation due to larger percentage of IPSec overhead for smaller packets. The degree of degradation in response time, throughput and

transaction rate was most significant with ESP or ESP-A applied, due to the additional time required for encryption. Overall response time for application doubled when AH was applied, tripled for ESP and almost quadrupled for ESP-A.

♦ *Case A — Response Time*

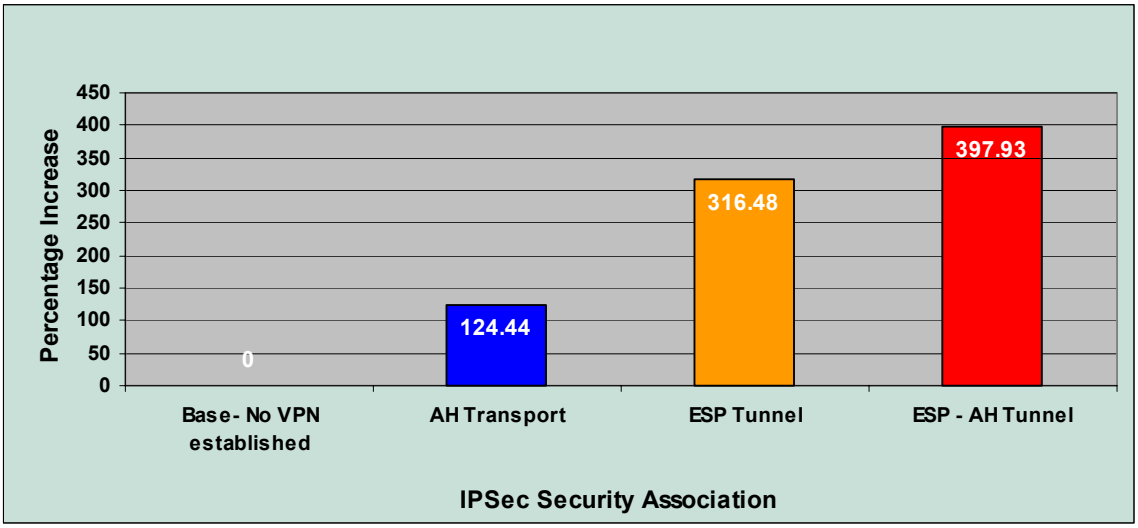


Figure 4.2 Case A — Response time

➔ Response time with an IPSec SA applied, on the average, increased 124% with AH only, 316% with ESP only and 398% with ESP-A applied.
➔ Most significant transaction types affected were multiple query-based transactions to a database and file transfers of small size.
➔ The IPSec SAs did not affect POP3 transactions, as POP3 is insensitive to delay.
➔ Response times grew to unacceptable levels (>200ms) for the IPSec SAs of ESP and ESP-A over a LAN connection.
➔ The increased response time makes it difficult for applications dependent upon the Transmission Control Protocol (TCP) and other connection oriented Transport Layer Protocols to sustain high a throughput.
➔ Increased response time was due to the additional processing time required by the VPN gateway to encrypt and or authenticate each the packet. For encryption, the larger the packet the longer this per packet processing time increased.

Response Time (Sec)	TRANSACTION TYPE											
IPSEC VPN Type	Credit Long	Database Long Transaction	POP3	File Send Short	File Send Small	Large File Send	Active Directory Log	Active Directory Replication	Exchange Server Read	Exchange Server Send	Exchange Sevrer Received	
Base- No VPN established	0.0010	0.0020	0.2000	0.1120	0.0120	1.0300	0.3064	10.3526	0.0210	0.0628	0.0210	
AH Transport	0.0050	0.0100	0.2000	0.2076	0.0292	1.8460	0.4586	17.5176	0.0310	0.0922	0.0310	
ESP Tunnel	0.0050	0.0110	0.2000	0.5022	0.0512	4.6560	1.2708	47.7736	0.0860	0.2580	0.0856	
ESP - AH Tunnel	0.0060	0.0140	0.2000	0.5836	0.0602	5.4700	1.5300	58.6276	0.1020	0.3060	0.1018	
Time Increase (%)												AVE
Base- No VPN established	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
AH Transport	400.00	400.00	0.00	85.36	143.33	79.22	49.67	69.21	47.62	46.82	47.62	124.44
ESP Tunnel	400.00	450.00	0.00	348.39	326.67	352.04	314.75	361.46	309.52	310.83	307.62	316.48
ESP - AH Tunnel	500.00	600.00	0.00	421.07	401.67	431.07	399.35	466.31	385.71	387.26	384.76	397.93

Figure 4.3 Case A — Response Time by Transaction

♦ *Case A — Throughput*

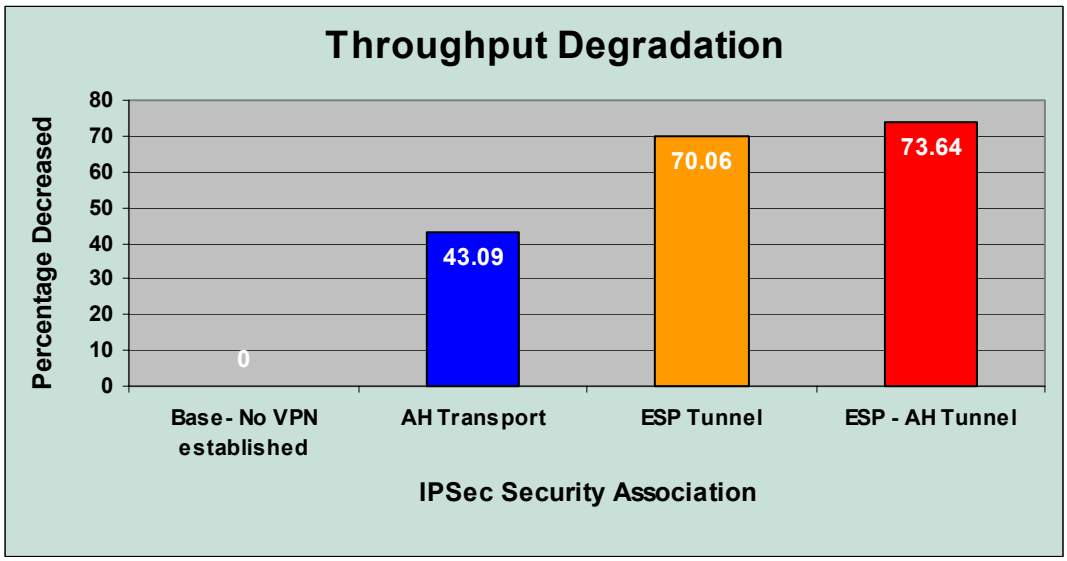


Figure 4.4 Case A — Throughput Degradation

➔ Throughput with an IPSec SA applied, on the average, decreased 43% with AH only, 70% with ESP only and 74% with ESP-A applied.
➔ The IPSec SAs did not affect POP3 transactions.

Throughput (Mbps)	TRANSACTION TYPE											
IPSEC VPN Type	Credit Long	Database Long Transaction	POP3	File Send Short	File Send Small	Large File Send	Active Directory Log	Active Directory Replication	Exchange Server Read	Exchange Server Send	Exchange Sevrer Received	
Base- No VPN established	0.8100	1.0702	0.0440	7.1328	3.2118	7.7700	0.1494	3.5922	1.3476	0.3630	0.3520	
AH Transport	0.1716	0.2444	0.0440	3.8548	1.3696	4.3330	0.1000	2.1228	0.9166	0.2462	0.2382	
ESP Tunnel	0.1662	0.2272	0.0440	1.5934	0.7808	1.7180	0.0362	0.7786	0.3286	0.0882	0.0854	
ESP - AH Tunnel	0.1270	0.1748	0.0440	1.3708	0.6640	1.4630	0.0300	0.6344	0.2770	0.0742	0.0720	
Percent Degradation												AVE
Base- No VPN established	0	0	0	0	0	0	0	0	0	0	0	0
AH Transport	78.81	77.16	0.00	45.96	57.36	44.23	33.07	40.91	31.98	32.18	32.33	43.09
ESP Tunnel	79.48	78.77	0.00	77.66	75.69	77.89	75.77	78.33	75.62	75.70	75.74	70.06
ESP - AH Tunnel	84.32	83.67	0.00	80.78	79.33	81.17	79.92	82.34	79.44	79.56	79.55	73.64

Figure 4.5 Case A — Throughput Degradation by Transaction

♦ *Case A — Transaction Rate*

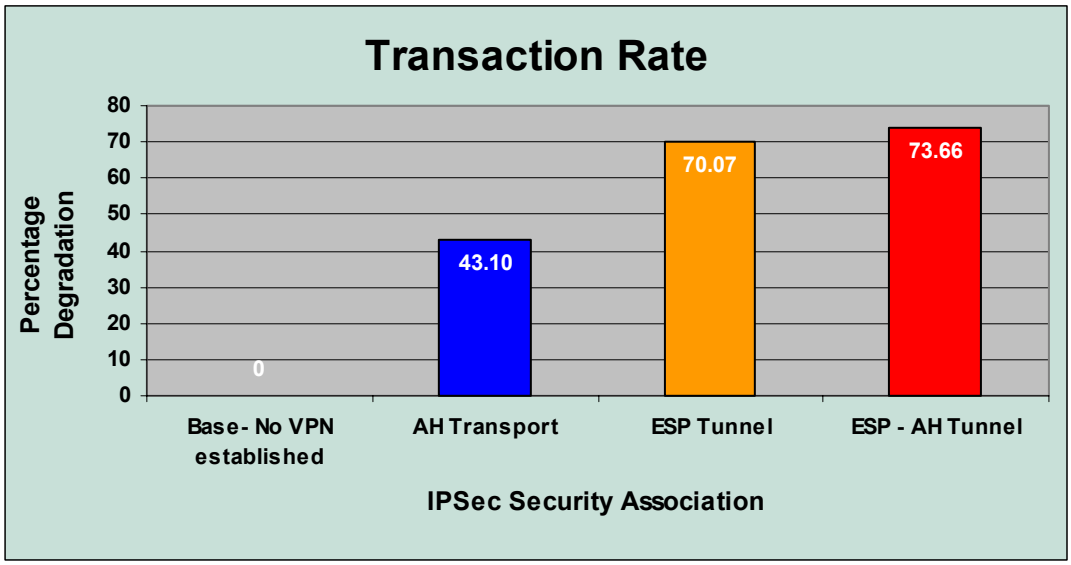


Figure 4.6 Case A — Transaction Rate

➔ Transaction Rate with an IPSec SA applied, on the average, decreased 43% with AH only, 70% with ESP only and 74% with ESP-A applied.
➔ Most significant transaction types affected were multiple query-based transactions to a database and file transfers of small size.
➔ The IPSec SAs did not affect POP3 transactions.

Transaction Rate (#/Sec)	TRANSACTION TYPE											
IPSEC VPN Type	Credit Long	Database Long Transaction	POP3	File Send Short	File Send Small	Large File Send	Active Directory Log	Active Directory Replication	Exchange Server Read	Exchange Server Send	Exchange Sevrer Received	
Base- No VPN established	1002.5926	444.3978	4.9974	8.9162	80.2828	0.9710	3.2630	0.0966	47.6384	15.9544	48.2586	
AH Transport	212.2364	101.4796	4.9954	4.8182	34.2294	0.5420	2.1808	0.0570	32.4092	10.8334	32.6396	
ESP Tunnel	205.6806	94.4442	4.9956	1.9916	19.5138	0.2150	0.7870	0.0210	11.6160	3.8754	11.6908	
ESP - AH Tunnel	157.2370	72.5996	4.9944	1.7134	16.5924	0.1830	0.6536	0.0170	9.7954	3.2694	9.8246	
Percent Degradation												AVE
Base- No VPN established	0	0	0	0	0	0	0	0	0	0	0	0
AH Transport	78.83	77.16	0.04	45.96	57.36	44.18	33.17	40.99	31.97	32.10	32.37	43.10
ESP Tunnel	79.49	78.75	0.04	77.66	75.69	77.86	75.88	78.26	75.62	75.71	75.77	70.07
ESP - AH Tunnel	84.32	83.66	0.06	80.78	79.33	81.15	79.97	82.40	79.44	79.51	79.64	73.66

Figure 4.7 Case A — Transaction Rate by Transaction Type

♦ *Case A — Measured Time*

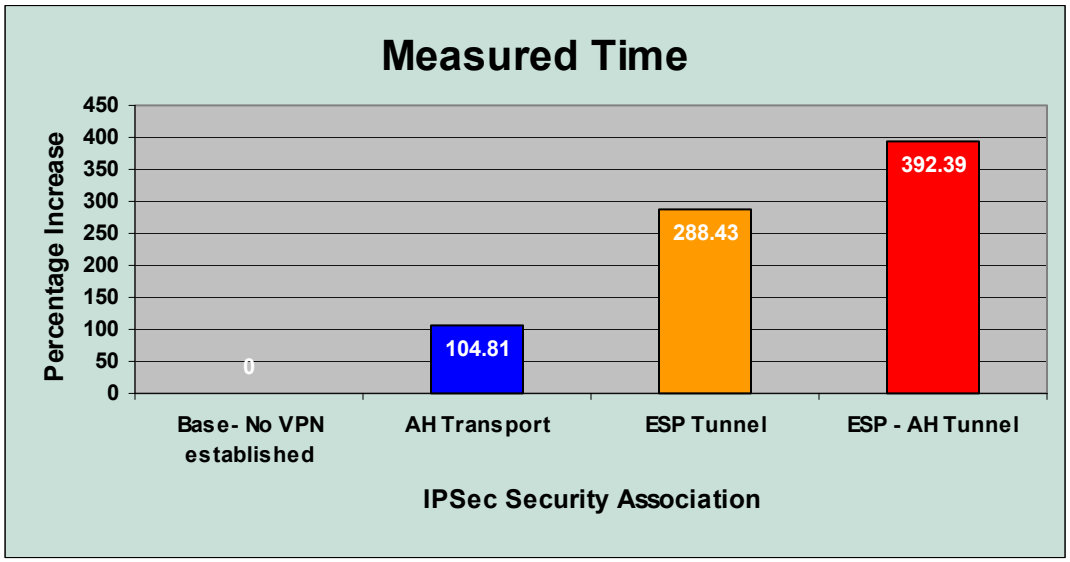


Figure 4.8 Case A — Measured Time

➔ Total measured time with an IPSec SA applied, on the average, increase 104% with AH only, 288% with ESP only and 392% with ESP-A applied.

➔ Increased measured time was due increased response time, decreased throughput as a result of the additional processing time required by the VPN gateway to encrypt and or authenticate each the packet.

Measured Time (sec)	TRANSACTION TYPE											
IPSEC VPN Type	Credit Long	Database Long Transaction	POP3	File Send Short	File Send Small	Large File Send	Active Directory Log	Active Directory Replication	Exchange Server Read	Exchange Server Send	Exchange Sevrer Received	
Base- No VPN established	2.4936	2.8128	50.0260	11.2156	2.4912	102.9580	22.9850	20.7046	25.7150	25.0716	25.9026	
AH Transport	11.7794	12.3178	50.0464	20.7548	2.9216	184.6340	34.3918	35.0344	37.7984	36.9228	38.2972	
ESP Tunnel	12.1548	13.2354	50.0426	50.2180	5.1252	465.6090	95.3108	95.5466	105.0264	103.2116	106.9276	
ESP - AH Tunnel	15.8996	17.2178	50.0558	58.3656	12.0540	546.9880	114.7398	117.2542	125.0578	122.3418	127.2344	
Time Increase (%)												AVE
Base- No VPN established	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
AH Transport	372.39	337.92	0.04	85.05	17.28	79.33	49.63	69.21	46.99	47.27	47.85	104.81
ESP Tunnel	387.44	370.54	0.03	347.75	105.73	352.23	314.67	361.48	308.42	311.67	312.81	288.43
ESP - AH Tunnel	537.62	512.12	0.06	420.40	383.86	431.27	399.19	466.32	386.32	387.97	391.20	392.39

Figure 4.9 Case A — Measured Time by Transaction Type

4.2.2 Case B — CMP Metrics for VPN Performance

Throughput, response and elapsed time measurements were calculated utilizing CMPMetrics for VPNs (CMPMetrics) and NetIQ's Performance Endpoint software between Host1 and Host2 over IPSEC SA. Five iterations of the test were conducted for each SA, and the results for the iterations were averaged together. In addition to throughput, response and elapsed time, CMPMetrics test also computes a relative value based on the performance results. The relative value was intended to provide users of CMPMetrics with a means to compare different VPN solutions. For the purpose of this study, the relative value only serves to highlight that a performance disparity exist between implementations. Actual performance characteristic values must be reviewed for actual comparisons.

4.2.2.1 Case B — Methodology

CMPMetrics utilizes Chariot scripts to emulate the interactions between computers on a live network. The execution process is similar to that of Case A, except the results are reported to CMPMetrics in lieu of Chariot.

The testing scenario under CMPMetrics consists of a combination of four separate tests. Test 1 & 2 are designed to run for approximately 1 minute, the duration for Test 3 & 4 is five seconds each. From the CMP Metrics for VPNs Performance description the tests are:

- *Test 1:* Host1 sends two files simultaneously to Host2. This test uses a modified Chariot's "File Send, Long Connection" (filesndl.scr) application script. Only one connection is established for the entire script. One file contains data from a newsgroup session, while the other is a graphics (.gif) file. Each file contains 100,000 Bytes of data. Because most VPNs compress encrypted data, the different filetypes test the VPN's compression and encryption capabilities: the news file is a text file that is easily compressed, while the binary graphics file is very difficult to compress. Compression and encryption affect CPU utilization on the device performing the VPN functions.

- *Test 2:* Host1 receives two files simultaneously from Host2. This test also uses a modified "File Receive, Long Connection" (filercvl.scr) application script and sends the same newsgroup and binary graphics data files. Each file contains 100,000 Bytes of data.

- *Test 3:* Host1 performs two simultaneous "small packet" inquiry transactions with Endpoint 2. This test uses a modified Chariot "Inquiry, Long Connection" (inquiryl.scr) application script with both the send and reply packet size set to 64 Bytes. One inquiry contains data from a newsgroup session, while the other is a binary graphics (.gif) file.

- *Test 4:* Host1 performs two simultaneous "large packet" inquiry transactions with Host2. This test also uses a modified "Inquiry, Long Connection" application script with both the send and reply packet size set to 512 Bytes. One inquiry contains data from a newsgroup session, while the other is a binary graphics (.gif) file.

The number of times the data is sent in each timing record for Sending File and Receiving File Tests was set to 10, while the for Large Packet and Small Packet Tests was 100.

4.2.2.2 Case B — Findings

The relative score of network performance generated by CMPMetrics is based on the throughput, response time and elapsed time of the test indicated a severe decrease (74-80%) in overall network performance when an IPSEC ESP or ESP-A was applied. A significant decrease of 24% was seen when AH was applied. The response time for smaller packets was seen to be more significant than that for larger files, yet the response time for both file sizes doubled the response time. In the case of ESP-A, small file size response time was seven time greater than that on the network without an IPSec SA applied. Due to the slow response times and additional packet overhead created throughput decreased almost 30% for AH and more than 75% for ESP and ESP-A. Table 4.9 summarizes the findings for Case B.

CMPMetrics	Relative Score	Throughput (Send) (Mbps)	Throughput (Rcv) (Mbps)	Response Time (Small Packet) (sec)	Response Time (Large Packet) (sec)	Elapsed Time (mm:ss)
Baseline	693.58	7.858	7.798	0.0011	0.00348	87.20
AH_Transport	527.22	5.534	5.508	0.0067	0.00866	85.60
ESP_Tunnel	177.7	1.824	1.832	0.0069	0.01568	89.20
ESP_AH_Tunnel	154.58	1.564	1.568	0.0094	0.01948	90.20
% Change from Baseline						
AH_Transport	23.99	29.57	29.37	510.91	148.85	-1.83
ESP_Tunnel	74.38	76.79	76.51	529.09	350.57	2.29
ESP_AH_Tunnel	77.71	80.10	79.89	750.91	459.77	3.44
Performance	DECREASED			INCREASED		

Figure 4.9 Case B — CMPMetrics Performance Summarization

Case B — Response Time

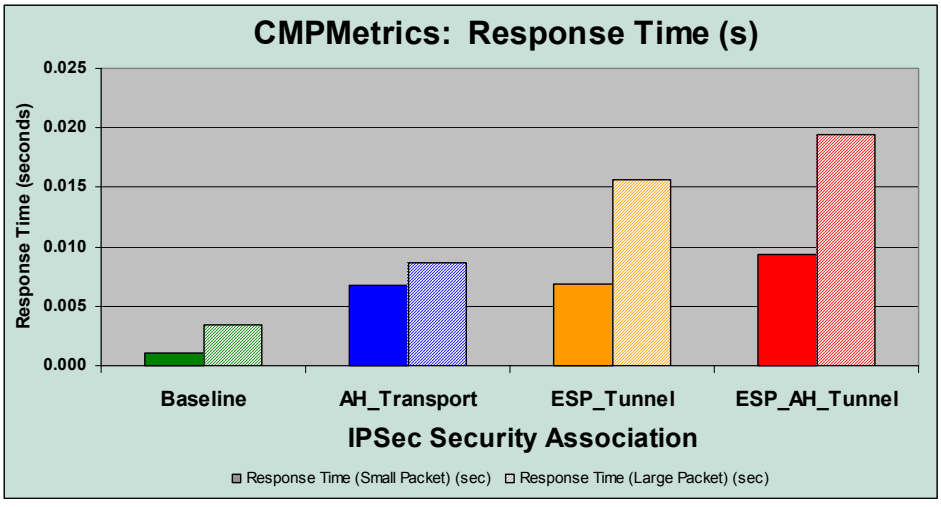


Figure 4.10 Case B — Response Time

➔ Response time for small packets dramatically increased by 510% with an AH IPSec SA, 529% for ESP, and 751% for ESP-A.
➔ Large packets also increased, but were not as sensitive to the response time; 149% for AH, 351% ESP and 460% for ESP-AH.

♦ *Case B — Throughput*

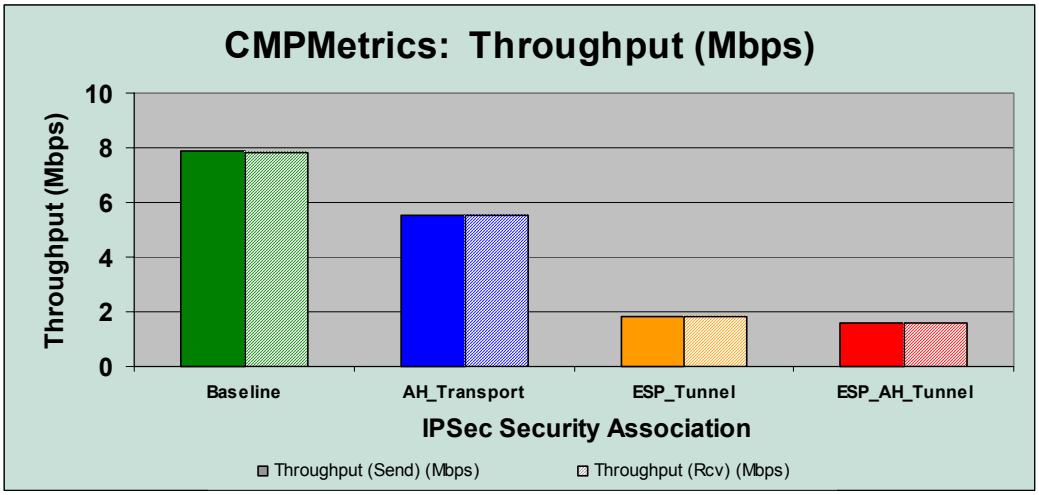


Figure 4.11 Case B — Throughput

➔ Throughput with an IPSec SA applied, on the average, decreased 29% with AH only, 77% with ESP only and 80% with ESP-A applied.

♦ *Case B — Elapsed Time*

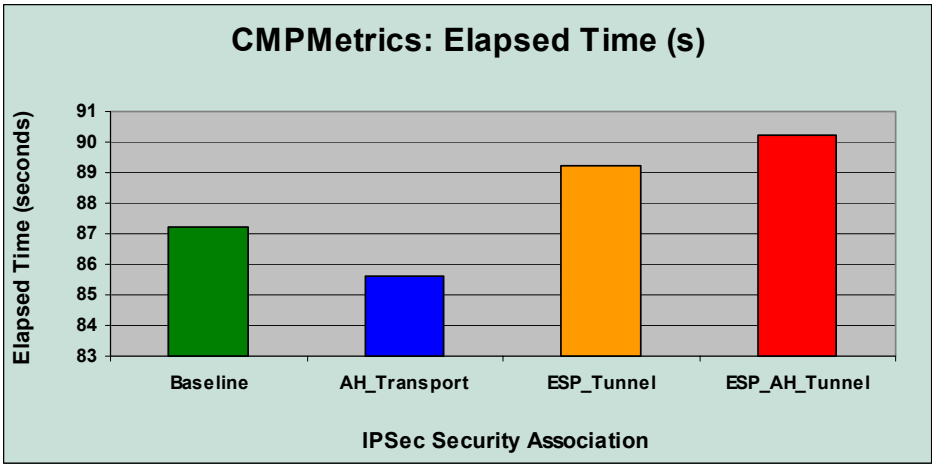


Figure 4.12 Case B — Elapsed Time

➔ Total elapsed time with an IPSec SA applied, on the average, decreased 2% with AH only, increased 2.9% with ESP only and 3.4% with ESP-A applied.

♦ *Case B — Relative Score*

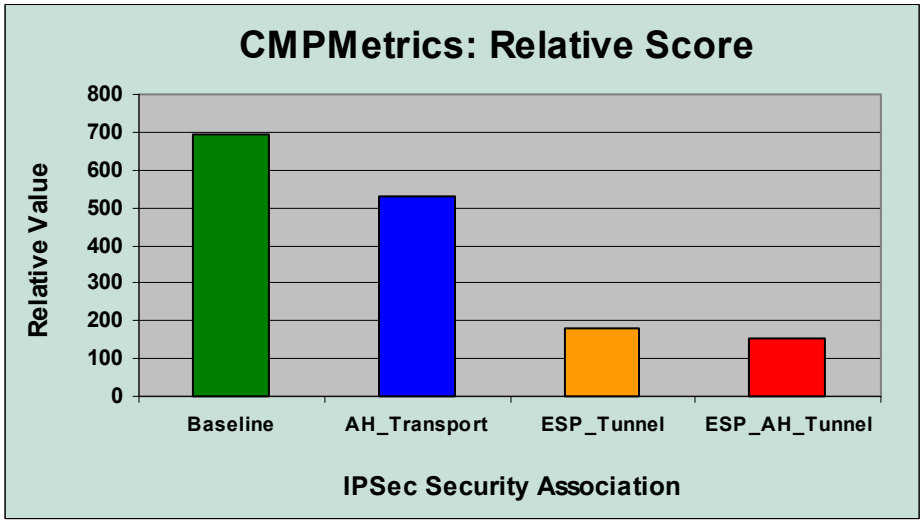


Figure 4.13 Relative Score

➔ The relative performance determined by CMPMetrics of the IPSec enable network in comparison to the baseline network (non-IPSec enabled) determined a degradation of 23% for AH, 74% for ESP, and 78% for ESP-A enable IPSec SAs.

4.2.3 Case C — UDP Flood

Utilizing a UDP packet injector utility called UDPFlood, Case C measures the nodal delay in a VPN device created by addition of an IPSec SA on packets as they are processed. The measured delays are then compared against the delay in a baseline network to determine the nodal processing delay caused by the addition of selected IPSec SAs.

4.2.3.1 Case C Methodology

VPN nodal delay is measured by using UDPFlood to send 200 UDP packets with a size 83, 1300 and 2500 bytes from Host1 to Host2 across IPSec SA1. The time between the UDP packets is measured on Host1, t_1, and again on SA1, t_2, after the packets have been processed and the IPSec SA applied. The difference between t_1 and t_2 is equal to the time required for Router1 to process the packet plus the propagation delay between Host1-Router1 and Router1-Console.

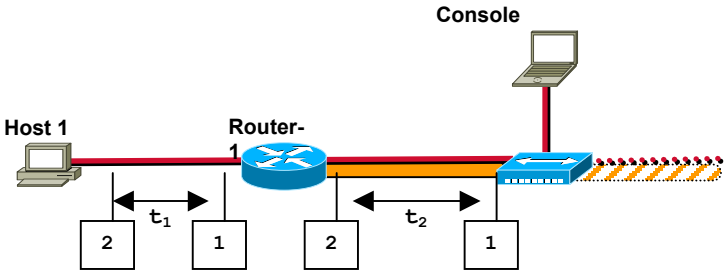


Figure 4.14 Measuring Nodal Delay

The cable distance between Host1-Router1 is 10M and the distance between Router1-Console is 3M, thus the total propagation delay is 65ns and becomes insignificant in determining the processing delay.

Velocity of transmission on copper media (V)= 2.01x10^8 m/s
Distance between Host1-Router1 $(d1)$= 10m
Distance between Router1-Console $(d2)$= 3m
Propagation delay between Host1-Console (Pd)=

$$Pd = \frac{d1}{V} + \frac{d2}{V} = \frac{10m}{2.01x10^8\, m/s} + \frac{3m}{2.01x10^8\, m/s} = 6.4675x10^{-8}\, s = 65ns$$

Therefore, t_2-t_1= processing delay.

To prevent any additional load on router1, an access list is placed on router2 to prevent any traffic destined for router1.

4.2.3.2 Case C — Findings.

The processing delay created by the application of an IPSec SA increased as the packet size increased, due to the additional calculations required by SHA-1 and 3DES needed to perform on large packets, the larger the packet, the longer the delay. As packets approach the

maximum transmission unit size, the more likely the chance for packet loss and much higher processing delay needed for both encryption and fragmentation. The test shows a high packet loss for packets of a size requiring the VPN device to fragment due to the additional bytes added from applying the IPSec SA causing the packet to exceed the MTU. We noticed an actual decrease in processing time of less than 1%, 7 µs, when AH was applied to packets of 83 and 1300 bytes. This decrease in time is believed to be attributed to the differences in clocks on the two computers recording the timing.

	Total Transmission Time Difference (ms)	Total Transmission Time Difference (%)	Internet Datagram Loss	Internet Datagram Loss (%)
AH (83)	-0.0009	-0.02	0.000000000	0
AH (1300)	-0.0067	-0.17	0.000000000	0
AH (2500)	0.1909	4.75	200.000000000	0
ESP (83)	0.0378	0.95	0.000000000	0
ESP (1300)	1.3278	33.04	0.000000000	0
ESP (2500)	7.2647	173.08	106.000000000	53
ESP-AH (83)	0.1351	3.38	0.000000000	0
ESP-AH (1300)	2.1037	52.05	0.000000000	0
ESP-AH (2500)	9.1041	223.07	124.000000000	62

Figure 4.15 Case C — UDP Flood Summary

♦ *Case C — Processing Time*

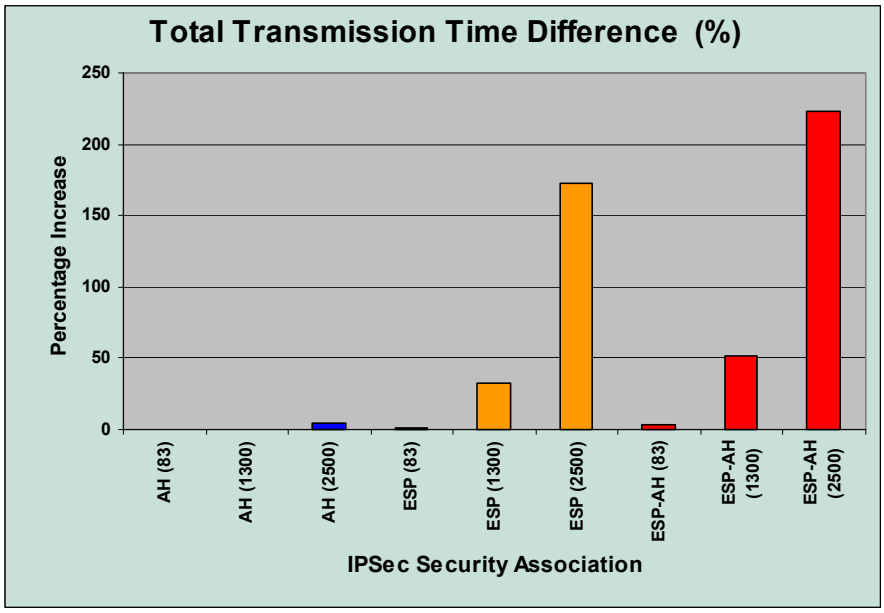


Figure 4.16 Case C — Total Transmission Time Difference

➔ Total elapsed time to process small packets with an IPSec SA was minimal, with a 3% time increase for an ESP-A IPSec SA.

- ➔ Semi-Large sized packets (1300 Bytes) saw a moderate jump in processing time- 33% for ESP and 52% ESP-A IPSec SAs.
- ➔ Performance was significantly degraded when maximized packets (1500 bytes) were encapsulated into an IPSec packet, as router1 was forced to fragment these packets to comply with the path maximum transmission size (PMTU). Overall processing time was increased 173% for ESP and 223% for ESP-A to handle the complete 2500 bytes IP datagram sent by Host1.
- ➔ The fragmentation observed in this test showed Host1 originally sending a UDP datagram with a size of 2500 bytes. Host1 was required to fragment that datagram into two packets. The first (P1) 1500 bytes and the second (P2) 1048 bytes. Router1, the VPN gateway, was forced to fragment P1, as the packet became oversized after applying the IPSec ESP SA into two packets, P1A and P1B. P1A was 1500 bytes in size and the fragment Router1 created, P1B, 60 bytes. P2, the packet fragment created by Host1, was processed as a normal IP packet by router1, whose size increased to 1092 after the IPSec ESP SA was applied. Packet sizes were slightly larger for when ESP-A was applied.
- ➔ AH was minimally affected by oversized datagrams - 4.75% increase in processing time.
- ➔ For both IPSec ESP and ESP-A SAs, CPU utilization on router1 reached 100% when processing the oversized datagrams of 2500 bytes, which in turn caused the router to begin to drop packets. CPU utilization averaged 93% utilization for packets of 1300 bytes.

♦ *Case C — Packet Loss*

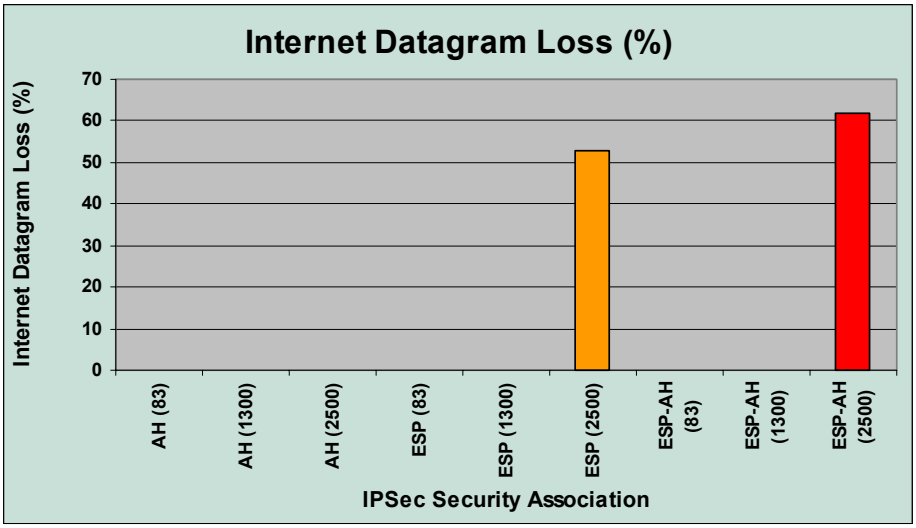


Figure 4.17 Case C — Internet Datagram Loss

- ➔ With an IPSec ESP SA applied, a 53% datagram loss was observed when the VPN device was required to fragment packets. With a ESP-A SA applied the datagrams loss grew to 63%.

4.3 Analysis of Results

4.3.1 Experimental Results

Experimental results from Cases A, B and C provide several points of discussion. All three cases show that greater SA complexity drives overhead up which lowers throughput. This in turn, raises response time and lowers the number of transactions per second. The overall effect is noticeable degradation of network performance.

Case A shows that when applications are sensitive to timely responses they incur a significant drop in throughput. Applications which generate small datagram are disproportionately impacted by the large overhead percentage per packet. As response times increase, network users perceive the drop in network performance. Most applications will timeout at any response time greater than 200ms. Case A clearly indicates that network performance varies inversely with increasing complexity of the specified Security Associations.

Case B demonstrated that VPN performance decreased (lower relative score) as the complexity of the SA increased. This supports the similar observations found in the Case A data. Analysis of Case B data showed that network transmission of smaller files suffered due to increasing overhead caused by increasing the complexity of the SA. The additional overhead consumed more of the devices processing time contributing to overall response delay. The analysis of Case B confirms the results from Case A.

Case C measured nodal delay. Case C analysis shows that more complex IPSec SA lead to greater CPU utilization. As datagram size increases they have a greater chance of suffering loss during VPN processing. Greater packet loss is due to maximum utilization of the CPU of the VPN device. Case C shows that IPSec has a direct impact on the CPU performance of the VPN device.

4.3.2 Overhead

The following discussion will outline the overhead in bytes per IP packet. The bytes per packet overhead directly affects the throughput of the transmission path. The added overhead decreases available space for input data. The overhead calculations for the AH header, ESP header, and ISAKMP header assume the smallest possible payloads . The figure below summarizes this information using a baseline of 40 bytes overhead for the IP and TCP encapsulation. The additional overhead is added on to the baseline.

	Baseline	AH Transport	ESP Tunnel	ESP-AH Tunnel
Overhead (Bytes per Packet)	40	64	82	94
Packet Size	% Overhead for TCP Packet			
60	66.67	106.67	136.67	156.67
120	33.33	53.33	68.33	78.33
180	22.22	35.56	45.56	52.22
240	16.67	26.67	34.17	39.17
300	13.33	21.33	27.33	31.33
360	11.11	17.78	22.78	26.11
420	9.52	15.24	19.52	22.38
480	8.33	13.33	17.08	19.58
540	7.41	11.85	15.19	17.41
600	6.67	10.67	13.67	15.67
660	6.06	9.70	12.42	14.24
720	5.56	8.89	11.39	13.06
780	5.13	8.21	10.51	12.05
840	4.76	7.62	9.76	11.19
900	4.44	7.11	9.11	10.44
960	4.17	6.67	8.54	9.79
1020	3.92	6.27	8.04	9.22
1080	3.70	5.93	7.59	8.70
1140	3.51	5.61	7.19	8.25
1200	3.33	5.33	6.83	7.83
1260	3.17	5.08	6.51	7.46
1320	3.03	4.85	6.21	7.12
1380	2.90	4.64	5.94	6.81
1440	2.78	4.44	5.69	6.53
1500	2.67	4.27	5.47	6.27

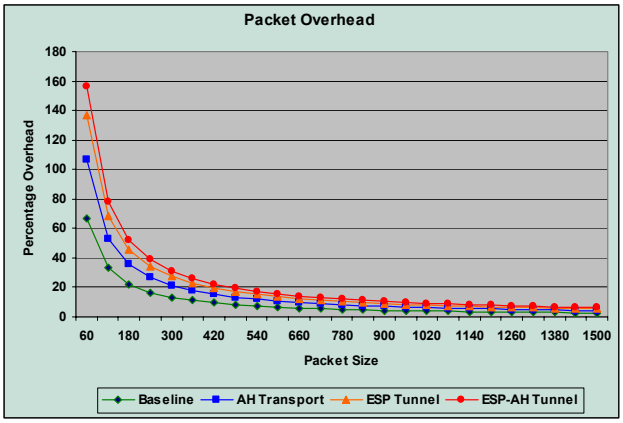


Figure 4.18 IPSec Overhead as a Percentage of Packet Size

Based on this table, some basic calculations can be utilized to calculate the percentage of overhead per packet. The basic methodology to calculate overhead as a percentage of packet size was as follows:

- A TELNET session was initiated and one character transmitted per packet. Each character equals a 1 byte datagram of information.
- TCP is the transmission protocol.
- TCP uses a 20 byte IP header + 20 byte TCP header + 1 byte data = 41 bytes.
- The AH_ESP_ah (authenticated encrypted packet encapsulated in authentication) adds 58 bytes.
- This original TELNET 41 byte packetized datagram is becomes 99 bytes due to the application of the AH_ESP_ah processing, resulting in a 241% increase in packet size.

To calculate a more accurate representation of the extreme cases, the figure below illustrates the maximum and minimum datagram payload with the corresponding packet size. The figure is based on the default 802.3 and Ethernet II MTU size and incorporates the two types of padding roles involved for encryption.

Maximum IP Datagram- ESP Only

	IP	ESP Hdr	ESP Payload	3DES Pad	ESP Pad	ESP Trailler	ESP Auth	Total Size
802.3	20	8	1446	0	0	2	12	1488
	20	8	1462	0	0	2		1492
Ethernet II	20	8	1454	0	0	2	12	1496
	20	8	1470	0	0	2		1500

Minimum IP Datagram- ESP Only

	IP	ESP Hdr	ESP Payload	3DES Pad	ESP Pad	ESP Trailler	ESP Auth	Total Size
802.3	20	8	1	4	1	2	12	48
	20	8	1	4	1	2		36
Ethernet II	20	8	1	4	1	2	12	48
	20	8	1	4	1	2		36

Maximum IP Datagram- AH plus ESP

	IP	AH Hdr	ESP Hdr	ESP Payload	3DES Pad	ESP Pad	ESP Trailler	ESP Auth	Total Size
802.3	20	24	8	1422	0	0	2	12	1488
	20	24	8	1438	0	0	2		1492
Ethernet II	20	24	8	1430	0	0	2	12	1496
	20	24	8	1446	0	0	2		1500

Minimum IP Datagram- AH Plus ESP

	IP	AH Hdr	ESP Hdr	ESP Payload	3DES Pad	ESP Pad	ESP Trailler	ESP Auth	Total Size
802.3	20	24	8	1	4	1	2	12	72
	20	24	8	1	4	1	2		60
Ethernet II	20	24	8	1	4	1	2	12	72
	20	24	8	1	4	1	2		60

ESP Encrypted Fields

AH Authenticated Fields + ESP Encrypted Fields

Assumptions:

* 3DES Encryption
* SHA-1 Authentication
* IP Options Not Used
* Calculations Based on MTU of Ethernet: Default MTU Size Ethernet II is 1500 Bytes and 802.3 is 1492 Bytes
* Maximum Packet Size is the largest size before the VPN device will Fragement the Packet

Figure 4.19 Minimum — Maximum Size for IPSec Packets

The ISAKMP overhead though possibly very large if using public certificates was considered insignificant. The basis for this conclusion is the ISAKMP main mode protocol structure is only utilized to setup the initial SA. Once established, policy will determine the requirement for frequency of transmission. This may be once a month, week, day, or some multiple of hours depending on the classification. Assumption for this study was that the ISAKMP SA would be renegotiated once every 3 days or 72 hours. With this assumption the ISAKMP main mode established overhead is insignificant or less than 0.00% of the packet throughput.

5 CONCLUSIONS, RECOMMENDATIONS AND EXTENSIONS

5.1 Conclusions

The implementation of DoD compliant VPNs will adversely affect network performance. The degree of adversity is dependent upon the hardware performance of the device performing the VPN functions of encryption and authentication, complexity of the security association and file size. As the complexity of the security association increases, the overall overhead of the generated packet increases as does the hardware requirements to process the packet.

This study indicates a direct correlation between the complexities of the security association and network performance. As additional requirements are included in the security association the overall packet size increases, as does the processing requirements to apply that security association to the packet. For example, when AH only is applied, the packet size increases by only 24 bytes. Along with the processing requirements needed for a larger packet, the VPN device must also calculate the Integrity Check Value. In a packet with ESP applied, the VPN device must now process not only a larger packet, it must determine what padding needs to be included and encrypt the packet.

As the size of a packet increases, the overall throughput increases since the ratio of overhead to data decreases. As the packet size begins to maximize, the CPU processing ability also reaches maximization (See Figure 5.1). At the point where CPU maximization occurs (Point X on Figure 5.1), packet loss begins to occur since the CPU can no longer handle the load presented. For connection-oriented protocols, such as TCP, congestion control mechanisms such as window sizing, retransmits and Nagles algorithm begin to have impacts at this point. Response time increases and the overall throughput of the connection decreases.

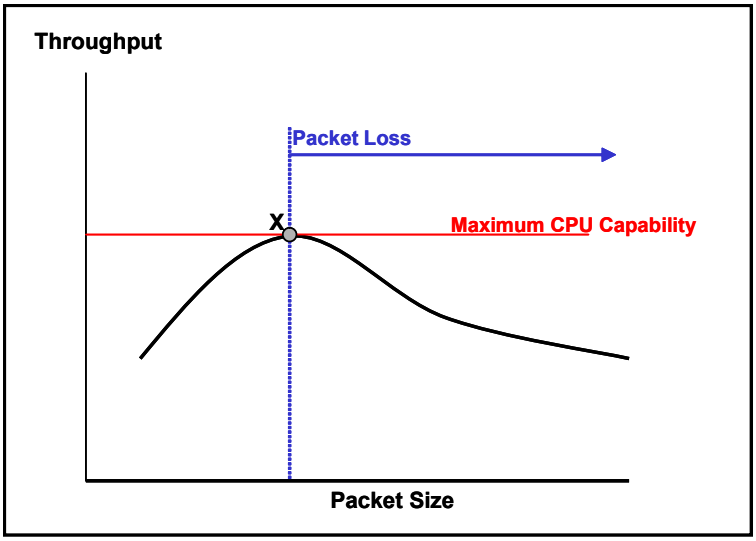


Figure 5.1 Throughput vs. Packet Size

For connectionless transport layer protocols, UDP, no congestion control is attempted resulting in lost packets and wasted throughput potential. The hardware's ability to perform the processing required maintaining a steady level of throughput decreases as the packet size increases.

A user on a network traversing a VPN will notice the degradation of networked application performance. This study found not only the degradation of the networks capability using precise measurement techniques to the millisecond level, but also delays a user would notice. The perception of poor network performance will only be compounded by already poor network links.

5.2 Recommendations

Prior to implementation of a VPN solution within the Army, a detailed review of the data traversing the VPN should be conducted to determine the actual need for the VPN. The results of such a review must indicate the need for data confidentiality for the data traversing the VPN. Once a need is determined, alternatives to implementing a VPN must be reviewed to ensure that other technologies do not satisfy the requirement at a much lower cost that the implementation of a VPN.

5.3 Extensions for Further Research

This study only attempts to highlight some performance impacts of VPNs implemented in accordance with DoD requirements; successful implementation of VPNs within the Army requires additional research. Additional studies determining the performance effects of layered SAs, maximum number of multiple connections before degradation on a single and layered SAs, and specific vendor implementation evaluations to include an assessment of the security compliance of that particular implementation is essential. This study focused on 3DES, additional research should also consider using AES and/or Skipjack IPSec implementations, in addition to comparing the performance of the three encryption techniques to determine which has a lower impact on network performance. As concluded in this study, the processing capabilities of the VPN device is a critical factor in the overall impact on network performance, thus a detailed review of software vs. hardware-based VPN functionality needs to be reviewed. Additional studies should also be performed to compare the technical and practical cost of VPNs in comparison to alternative confidentiality technologies and under which cases one is a preferred solution to the other from a network performance perspective.

REFERENCES

[1] Bruce Perlmutter, *Virtual Private Networking, A View From the Trenches*, (Upper Saddle River, NJ: Prentice Hall PTR, 2000), 27-29.

[2] USAISEC Security Engineering Team*, Army Virtual Private Networking (VPN) Architecture and Implementation Guide (DRAFT)* (Fort Huachuca, AZ: U.S. Army Information Systems Engineering Command, June 2001), 3-4.

[3] USAISEC, 4.

[4] *Common Criteria for Information Technology Security Evaluation, CC 2.1*, (Washington DC, US Government, August 1999), 1.

[5] National Security Agency*, A Goal VPN Protection Profile for Protecting Sensitive Information, Release 2.0,* (Washington, DC: National Security Agency, 10 July 2000), ii.

[6] NSA, iv.

[7] USAISEC, 7.

[8] Perlmutter, 106.

[9] S. Kent and R. Atkinson, "RFC 2401, Security Architecture for the Internet Protocol," (Internet Engineering Task Force, November 1998), 4.

[10] ibid., 8.

[11] ibid., 11.

[12] Ruixi Yuan and W. Timothy Stayer, *Virtual Private Networks, Technologies and Solutions*, (Boston: Addison-Wesley, 2001), 79.

[13] Carlton R. Davis, *IPSec, Securing VPNs*, (New York: Osborne McGraw Hill, RSA Press, 2001), 187.

[14] S. Kent and R. Atkinson, "RFC 2402 IP Authentication Header," (Internet Engineering Task Force, November 1998), 3-5.

[15] Davis, 195 .

[16] S. Kent and R. Atkinson, "RFC 2406, IP Encapsulating Security Payload," (Internet Engineering Task Force, November 1998), 3-7.

[17] Davis, 215.

[18] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "RFC 2408, Internet Security Association and Key Management Protocol," (Internet Engineering Task Force, November 1998), 8.

[19] D. Maughan, et al, 21-25.

[20] ibid., 25-45.

[21] D. Harkin, and D. Carrel, "RFC 2409, The Internet Key Exchange," (Internet Engineering Task Force, November 1998), 2.

[22] Cooperative Association for Internet Data Analysis (CADIA), http://www.caida.org.

[23] Opnix, Inc's Internet Traffic Report, http://www.internettrafficreport.com.

[24] Message and Application Scripts (PDF), NetIQ Corporation, 31 Mar 02, Avail at http://www.netiq.com/support/chr/as/documentation.asp.